City-Scale Traffic Animation Using Statistical Learning and Metamodel-Based Optimization

WEIZI LI, DAVID WOLINSKI, and MING C. LIN, University of North Carolina at Chapel Hill

Fig. 1. 2D animation (left) and visualization (center) of reconstructed traffic in virtual San Francisco with 3D vehicle flows (right) using our method.

Rapid urbanization and increasing traffic have caused severe social, economic, and environmental problems in metropolitan areas worldwide. Traffic reconstruction and visualization using existing traffic data can provide novel tools for vehicle navigation and routing, congestion analysis, and traffic management. While traditional data collection methods are becoming increasingly common (e.g. using in-road sensors), GPS devices are also becoming ubiquitous. In this paper, we address the problem of traffic reconstruction, visualization, and animation using mobile vehicle data (i.e. GPS traces). We first conduct city-scale traffic reconstruction using statistical learning on mobile vehicle data for traffic animation and visualization, and then dynamically complete missing data using metamodel-based simulation optimization in areas of insufficient data coverage. We evaluate our approach quantitatively and qualitatively, and demonstrate our results with 2D visualization of citywide traffic, as well as 2D and 3D animation of reconstructed traffic in virtual environments.

CCS Concepts: • **Computing methodologies** → **Machine learning**; *Learning in probabilistic graphical models*; **Animation**; *Physical simulation*;

Additional Key Words and Phrases: Traffic Reconstruction, Traffic Simulation, GPS data, Statistical Learning, Metamodel-based Optimization

ACM Reference format:
Weizi Li, David Wolinski, and Ming C. Lin. 2017. City-Scale Traffic Animation Using Statistical Learning and Metamodel-Based Optimization. *ACM Trans. Graph.* 36, 6, Article 200 (November 2017), 12 pages.
https://doi.org/10.1145/3130800.3130847

1 INTRODUCTION

Rapid urbanization and increasing traffic have a serious social, economic, and environmental impact on metropolitan areas worldwide. The ever-present congestion restricts the accessibility of a city and affects the mobility of its inhabitants [Schrank et al. 2015]. Many efforts have been invested in digitalizing and visualizing urban environments, e.g. software tools like Google Maps and Virtual Earth. As new technologies like VR systems [MIT 2011; TAC 2017; Wang et al. 2005] and self-driving cars emerge, there is an increasing demand to incorporate realistic traffic flows into virtualized cities. The potential applications range from virtual tourism, networked gaming, navigation services, urban design, to training of autonomous vehicles. The ability to reconstruct city-scale traffic from mobile sensor data can enable the visualization and animation of realistic real-world traffic conditions thus contributes to those applications.

Traditional traffic data collection methods (e.g. in-road sensors like loop detectors and video cameras) are very costly; new and cheaper data sources such as GPS devices are becoming increasingly ubiquitous. Especially, taxicabs and shared ride services (e.g. Uber and Lyft) are systematically equipping their car fleets with these devices. As a result, GPS traces are part of the most promising data sources to estimate citywide traffic conditions attributing to its broad coverage[1]. However, besides the inherent noise, GPS data usually contains a low-sampling rate; i.e., the time lapse between two consecutive sampled points is large (e.g. >30 seconds). The result could be multiple paths for connecting two consecutive points in an urban environment. In addition, GPS data exhibits spatial-temporal sparsity; i.e., the data is scarce in certain areas (e.g. suburbs) and time periods (e.g. early-morning hours), which makes city-scale estimation very difficult.

While it is already challenging to reconstruct traffic conditions in areas where GPS data is available, in order to reconstruct citywide traffic, data completion is needed in areas without GPS data. In theory, the local traffic in these areas can be approximated using microscopic traffic simulation, which capture interactions among individual vehicles. However, it is critical to ensure consistency of traffic flows on the boundary of areas without GPS data and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.
0730-0301/2017/11-ART200 $15.00
https://doi.org/10.1145/3130800.3130847

[1] Another ubiquitous data source is the cellular network. However, it is usually inadequate for estimating traffic states alone [Wu et al. 2015].

areas with GPS data. This indicates that microscopic traffic simulations must be *dynamically* tuned to ensure matching flows with the reconstructed traffic in regions with heavy and complete GPS coverage.

Reconstructing city-scale traffic dynamics using mobile vehicle data (i.e. GPS traces) presents a number of challenges: (1) processing the available data, (2) coping with insufficient data coverage, and (3) reconstructing local traffic flows that are consistent with the global traffic dynamics at large temporal and spatial scales. In order to address these issues, we propose a systematic approach that takes a GIS map (processed using [Wilkie et al. 2012]) and GPS data (collected from taxicabs in San Francisco in 2008 [Piorkowski et al. 2009]) as input, and reconstruct the city-scale traffic using a two-phase process. In the first phase of our system, we reconstruct and progressively refine flow conditions on individual road segments from the sparse GPS data using statistical learning combined with optimization, *map-matching* [Quddus and Washington 2015], and *travel-time estimation* [Li et al. 2017] techniques. In the second phase of our system, we use a metamodel-based simulation optimization to efficiently refine the reconstructed results from the previous phase, along with a microscopic simulator [Krajzewicz et al. 2012] to dynamically interpolate missing data. To ensure that the reconstructed traffic is correct, we fine tune the simulation with respect to city-wide boundary (traffic) constraints and the reconstructed traffic flow from the first phase; this objective is enforced through an error approximation of the traffic flow computed by our novel metamodel-based formulation. In summary, we address the problem of learning-based traffic animation and visualization using GPS data. Our contributions are as follows:

- City-scale traffic reconstruction using statistical learning on GPS data for traffic animation and visualization;
- Dynamic data completion using metamodel-based simulation optimization in areas of insufficient data coverage.

The rest of the paper is structured as follows. We survey related work in traffic reconstruction and simulation in Section 2. A general overview is provided in Section 3. We detail our approach in Sections 4 and 5. We present results for reconstruction and simulation, as well as system validation and application demonstration in Section 6. Finally, we conclude and discuss future work in Section 7.

2 RELATED WORK

Modeling of urban environments has received considerable attention in recent years. Many systems have been developed to describe a variety of their aspects (layout, vehicle traffic, pedestrian motion, etc.) [Garcia-Dorado et al. 2017; Li et al. 2016; Musialski et al. 2013; Thomas and Donikian 2000; Willemsen et al. 2006]. In terms of vehicle motion, a number of driving simulators and vehicle behavioral models have been proposed [MIT 2011; TAC 2017; Wang et al. 2005], with several designed for virtual reality systems [Bayarri et al. 1996; Kuhl et al. 1995; Wang et al. 2005]. Considerable advances have also been made in the development of increasingly open-world video games (e.g. Grand Theft Auto, Burnout Paradise, and Watch Dogs) and visualization of traffic [Andrienko and Andrienko 2006; Ferreira et al. 2013; Wang et al. 2014].

Traffic simulation has received renewed interest in the past decade, for example, with the introduced concept of "virtualized traffic" [Sewall et al. 2011a; van den Berg et al. 2009], and the increasing efforts towards modeling of realistic traffic flows [Wilkie et al. 2015]. Some recent works include an extension of macroscopic models to generate detailed animations of traffic flows [Sewall et al. 2010], and their integration with existing microscopic models to produce traffic animation on urban road networks [Sewall et al. 2011b]. [Shen and Jin 2012] and [Mao et al. 2015] have improved existing microscopic models to produce believable traffic animations. Techniques from texture synthesis have been adopted to enhance the visual quality of traffic flows [Chao et al. 2017]; characterization of heterogeneous vehicle types [Lin et al. 2016] and driver personalities [Lu et al. 2014] have been explored. Real-world data has also been used to calibrate simulated traffic flows: recent works, including [van den Berg et al. 2009], [Sewall et al. 2011a], and [Wilkie et al. 2013], all have explored in-road sensors to reconstruct the traffic flow; [Chao et al. 2013] acquired individual vehicle characteristics from video cameras; [Bi et al. 2016] adopted a data-driven method to enrich the lane-changing behaviors of traffic simulations. Finally, [Garcia-Dorado et al. 2014] provided users with the flexibility to assign desired vehicular behaviors to a road network.

Data-driven modeling has been studied by several recent works for crowds [Charalambous and Chrysanthou 2014; Ju et al. 2010; Lee et al. 2007; Lerner et al. 2007], which like traffic, are often modeled using multi-agent simulation. There are however two main distinctions to be noted. The first is the *problem of scale*. While data-driven crowds are often limited to a few hundreds of individuals, city-scale traffic reconstruction concerns tens of thousands of vehicles. The second distinction is the nature of the data: in data-driven crowds, at least several positions and joint angles are recorded per second for each individual, while mobile traffic data is often very sparse and individual trajectories are not known and cannot be assumed.

Traffic reconstruction has drawn much attention in the field of traffic engineering [Kachroo and Sastry 2016]. In order to achieve high accuracy, multiple data sources and traffic simulation models have been investigated [Li et al. 2014; Perttunen et al. 2015; Work et al. 2010]. While significant results have been achieved, these methods along with other projects such as Mobile Century [Herrera et al. 2010] are largely restricted to highway segments with lengths of a few kilometers. In order to expand reconstructions to arterial roads and surface streets, recent studies [Castro et al. 2012; Kong et al. 2013; Zhang et al. 2013] have adopted GPS data, which presents a number of challenges. Primarily, due to the uncertainty and sparsity of the data, several processing steps are required and are usually conducted in tandem.

The first step in the pipeline is *map-matching*, which addresses off-the-road GPS points and infers the truly traversed path connecting them. One of the commonly used techniques for connecting *low-sampling-rate* points is to take the shortest-distance path [Hunter et al. 2014; Quddus and Washington 2015]. Unfortunately, this criterion produces biased results in congested environments where the shortest-distance path is no longer the shortest *travel-time* path, which drivers would prefer [Hunter 2014; Li et al. 2017; Tang et al.

2016]. However, finding the shortest travel-time path requires accurate estimations of traffic conditions, which are difficult to obtain. The second step, *travel-time estimation*, tries to distribute the aggregate travel time to individual road segments of a map-matched path. To list a few examples, [Hellinga et al. 2008] developed an analytical solution based on empirical observations of traffic patterns; [Rahmani et al. 2015] took a non-parametric approach and adopt a kernel-based estimator; other approaches are based on probabilistic frameworks [Herring et al. 2010; Hunter 2014]. While encouraging improvements have been made, these methods suffer from the inherent limitations of *map-matching* under the sequential process. Especially, in a congested network, the aggregate travel time is likely to be assigned to a wrong set of road segments given the shortest-distance criterion. The main difference between our *travel-time estimation* technique and previous works such as [Hunter 2014; Li et al. 2017; Tang et al. 2016] is that we take an iterative rather than a sequential perspective so that the errors generated during each step get gradually attenuated.

Our work differs from the previous efforts in several regards. First, traffic flow reconstruction on arterial roads presents new challenges over that on highways and major roads, where in-road sensors are available. The simulation techniques used in recent works, such as [van den Berg et al. 2009; Wilkie et al. 2013], depend on relatively accurate data and certain macroscopic traffic assumptions. However, GPS data is inherently noisy and the macroscopic traffic assumptions have been shown to break down on arterial roads [Kong et al. 2013]. Our framework addresses these issues by conducting statistical learning on mobile data and reconstructs traffic in regions which have a rich data coverage. Second, while efficient and large-scale traffic simulation techniques have been introduced [Sewall et al. 2011b, 2010], these techniques are not designed to ensure simulation fidelity with respect to reconstructed traffic flows. Our system dynamically completes the reconstruction in areas where mobile data is insufficient or missing. This is achieved by using metamodel-based optimization to satisfy city-wide boundary constraints and to minimize the error differences to the reconstructed traffic.

3 OVERVIEW

In this section, we first give an overview of our approach and then define the notations used in this paper.

3.1 System Framework

In order to achieve city-scale traffic visualization and animation, our system consists of two main phases: (1) an initial traffic reconstruction phase (center-left column in Figure 2, and Section 4) that estimates *initial* traffic flows on all road segments in the target area, and (2) a dynamic data completion phase (center-right column in Figure 2, and Section 5) that yields a *dynamic* and much more accurate reconstructed traffic efficiently at the city scale.

For the *Initial Traffic Reconstruction (Section 4)* phase, one of the main objectives is to reconstruct road-segment flows from the input data. As mentioned before, this problem presents two challenges, which we will address in order.

The first one is the sparseness of GPS data, where consecutive data-points are on average 60 seconds apart. To cope with this issue, we embed *map-matching* [Quddus and Washington 2015] and *travel-time estimation* [Li et al. 2017] into an iterative process (top and bottom boxes, center-left column in Figure 2, and Section 4.2), where the output of one is treated as the input of the other and vice-versa. Overall, this process progressively refines an estimation of travel times on road segments until convergence. This iterative process is initialized with a starting estimation obtained through a "naïve" optimization (left box, center-left column in Figure 2, and Section 4.1). After the iterative process, we obtain flows for all road segments in areas that have GPS data coverage.

The second challenge is the uneven coverage of GPS data. In order to reconstruct traffic flows in areas with little to no data, we perform a bilevel optimization (right box, center-left column in Figure 2, and Section 4.3). The main principle of this phase is to use *existing data* (either outdated traffic data or estimated results) that is complete but not very precise, and update it using *new information* (i.e. the reconstruction from the previous phase). As a result of this phase, we obtain flows for road segments in areas that have no GPS data coverage. Note however that in these areas, flows are only accurate at a large scale but not at the scale of individual road segments.

For the *Dynamic Data Completion (Section 5)* phase, we have obtained estimated flows from the previous phase that are only accurate at a large scale, but not necessarily at the road-segment scale: road-segment flows are accurate in areas with GPS data coverage, but not in areas without. Additionally, this reconstruction is static and does not account for interactions among individual cars, which are pronounced in arterial roads and have a profound impact on the overall traffic dynamics at the city scale. In order to address the missing data issue and unaccounted local interactions, we use a microscopic traffic simulator (i.e. SUMO [Krajzewicz et al. 2012]).

A simulation given by such an algorithm in a given area that does not have GPS data coverage might not however, respect boundary conditions and previously reconstructed traffic in data-rich regions "out of the box". To address these issues, we allow the algorithm to change the "turning ratios" at intersections (i.e. the parameters deciding how likely a car is to take a given exit when arriving at a traffic intersection). Changing these turning ratios effectively alters the simulated flows both for the road segments and for the overall traffic. The objective is then to change these turning ratios such that flows simulated at the boundaries correspond to the reconstructed traffic from GPS data. The simulated road-segment flows are then coherent with the overall traffic (thereby also correcting what has been inferred from *existing data* used in the bilevel optimization substep of traffic reconstruction).

In this form, the optimization of turning ratios (bottom and top boxes, center-right column in Figure 2, and Section 5.1) would be very costly because of the need to use the microscopic simulator in the cost function. Consequently, we accelerate this optimization step by approximating the cost function with a *metamodel* [Osorio et al. 2015] (top box, center-right column in Figure 2), thus making this optimization problem more tractable. Finally, we initialize the optimization scheme randomly (left box, center-right column in Figure 2, and Section 5.2), and use the resulting turning ratios of the optimization in the simulator to get the final results (right box, center-right column in Figure 2, and Section 5.2).

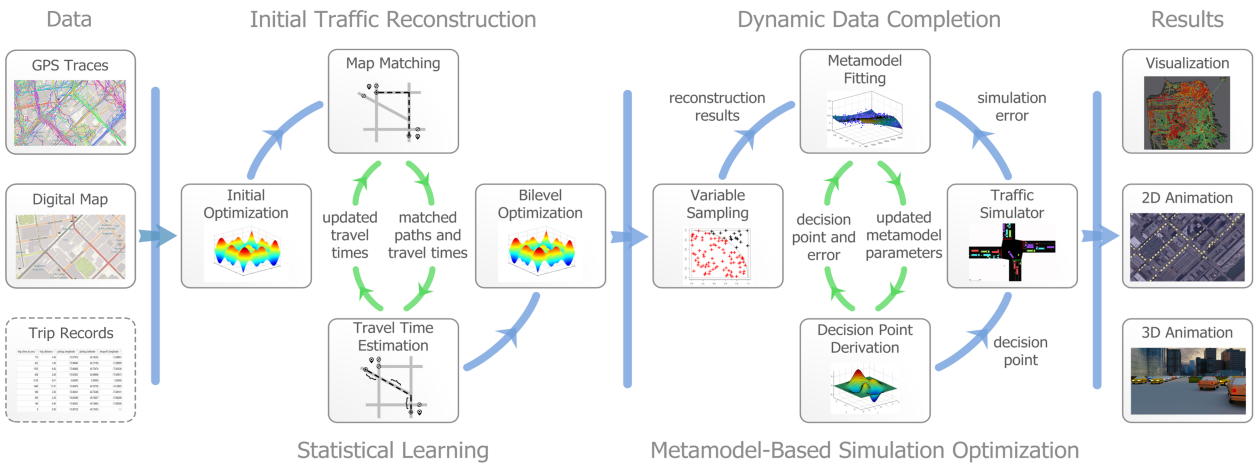

Fig. 2. The systematic view of our framework. Trip records are optional as they can be inferred from GPS traces on a digital map.

3.2 Notations

A road network is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which the nodes $\mathcal{V}$ and edges $\mathcal{E}$ respectively represent intersections (or terminal points), and road segments. Additionally, we denote a path on the road network as a set of road segments $k \in \mathcal{K}$ where $\mathcal{K}$ is the set of all paths.

Geographically, a city can be divided into many traffic analysis zones (TAZs) based on socio-economic data. The centroids of these TAZs are considered as traffic-flow *origins* and *destinations*. The sets of origins and destinations are respectively denoted as $\mathcal{O} \subseteq \mathcal{V}$ and $\mathcal{D} \subseteq \mathcal{V}$. The traffic flow is considered to take place between origin-destination pairs (OD pairs), and the average flow from $o \in \mathcal{O}$ to $d \in \mathcal{D}$ during a certain time interval is noted u_{od} (in accordance with traffic engineering literature, this flow is the average number of vehicles). With $\mathcal{K}_{od}$ as the set of paths connecting o and d (and $\mathcal{K}$ the set of all paths), the flow on path $k \in \mathcal{K}_{od}$ is $u(k)$. The flow inside an OD pair is a summation of flows on all paths in this pair:

$$u_{od} = \sum_{k \in \mathcal{K}_{od}} u(k),\ u(k) \geq 0, \forall o \in \mathcal{O}, d \in \mathcal{D}. \tag{1}$$

The flow $f(e)$ on a road segment $e \in \mathcal{E}$, is defined as the sum of all path flows traversing e:

$$f(e) = \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \sum_{k \in \mathcal{K}_{od}} \delta_e^k u(k), \tag{2}$$

where δ_e^k is 1 if path k contains road e, and 0 otherwise. This means that $\forall e \in \mathcal{E}, \forall k \in \mathcal{K}_{od}$, if $e \in k$, then e contributes a certain portion $p_{e,od}$ of u_{od}. Thus, by arranging the flows of all OD pairs in a vector $\mathbf{u} = [u_{od}]^\intercal_{(o,d) \in \mathcal{O} \times \mathcal{D}}$, and the contributed flow portions of a road segment e as $\mathbf{P}_e = [p_{e,od}]_{(o,d) \in \mathcal{O} \times \mathcal{D}}$, we have the following relation:

$$f(e) = \mathbf{P}_e \mathbf{u}. \tag{3}$$

With all road-segment flows in a network $\mathbf{f}_\mathcal{E} = [f(e)]^\intercal_{e \in \mathcal{E}}$ and the portions of all road segments $\mathbf{P}_\mathcal{E} = [\mathbf{P}_e]^\intercal_{e \in \mathcal{E}}$, we obtain the general relation between flows in road segments and OD pairs:

$$\mathbf{f}_\mathcal{E} = \mathbf{P}_\mathcal{E} \mathbf{u}, \tag{4}$$

with $\mathbf{P}_\mathcal{E}$ termed *assignment matrix*.

Other notations used in this work are as follows:

- $\mathcal{S}$ denotes the GPS data, then $\forall s \in \mathcal{S}$ contains a longitude, a latitude, and a timestamp; additionally, a pair of successive GPS data points can be noted $(s_1, s_2) \in \mathcal{S}_p$, where $\mathcal{S}_p$ would then be the set of all pairs of consecutive GPS data points; finally, $\mathcal{K}_{s_1,s_2}$ is the set of all paths connecting s_1 to s_2.
- t denotes travel time; for instance, $t(e)$ is the travel time on a road segment $e \in \mathcal{E}$, $t(k) = \sum_{e \in k} t(e)$ is the travel time on a path $k \in \mathcal{K}$, and $t(s_1, s_2)$ is the travel time between a pair of two consecutive GPS data points $(s_1, s_2) \in \mathcal{S}_p$ given by their timestamps.

Next, we describe the main phases of our framework in detail.

4 CITY-SCALE TRAFFIC RECONSTRUCTION

There are three steps in the *initial traffic reconstruction* phase: (1) initial estimation of travel times on road segments, (2) iterative refinement of these travel times, and (3) bilevel optimization to fill data-lacking areas.

4.1 Initial Estimation

For the initial estimation of travel times, we use Wardrop's principle [Wardrop 1952], which states that traffic will arrange itself in congested networks such that no vehicle can reduce its travel cost by switching routes. This state is termed *user equilibrium* and is a result of every user non-collaboratively attempting to minimize their travel times. While the actual traffic may not necessarily form a *user equilibrium*, it serves as an approximation to real-world traffic [Hato et al. 1999] and motivates the generation of the constraints in our optimization.

Following this principle, for a pair of two consecutive GPS data points $(s_1, s_2) \in \mathcal{S}_p$, the travel time between them $t(s_1, t_2)$ should be the minimum travel time on all paths connecting these two points: $\forall k \in \mathcal{K}_{s_1,s_2}, t(k) \geq t(s_1, s_2)$. Thus, travel times on road segments

should satisfy the following "Wardrop" constraints $\mathcal{W}$:

$$\mathcal{W} = \{t(k) \geq t(s_1, s_2)\}_{\forall (s_1,s_2) \in \mathcal{S}_p,\ \forall k \in \mathcal{K}_{s_1,s_2}}. \tag{5}$$

Additionally, we bound travel times on road segments $e \in \mathcal{E}$ by $t_{min}(e)$ and $t_{max}(e)$, respectively representing free-flow travel time (at 120% of the road segment's speed limit[2]) and travel time at jam density (at $0.5m.s^{-1}$).

In order to derive a solution with respect to typical traffic patterns, we pose a regularization term, $\mathcal{R}$, on $\mathbf{t}_\mathcal{E} = [t(e)]^\intercal_{e\in\mathcal{E}}$ to model the correlation in traffic patterns of road segments in close proximity [Sheffi 1985; Wang et al. 2016; Zheng et al. 2011]. $\mathcal{R}$ is formed as a 2-dimensional fused Lasso penalty [Tibshirani et al. 2005] in which each row represents a pair of road segments connecting at a node. The entries corresponding to the pair of road segments are set to 1 and -1, accordingly. As an example, the $\mathcal{R}$ of a three-way intersection with road segments $\{e_1, e_2, e_3\}$ takes the form: $\mathcal{R} = \{[1,1,0]^\intercal, [-1,0,1]^\intercal, [0,-1,-1]^\intercal\}$. We set up $\mathcal{R}$ by enumerating all pairs of road segments at each $v \in \mathcal{V}$. As a result, $\mathcal{R} \in \mathbb{R}^{m\times n}$, where $m = \sum_{v\in\mathcal{V}} \binom{deg(v)}{2}$, $n = |\mathcal{E}|$, and $deg(v)$ is the total degree of v. The initial estimation $\bar{\mathbf{t}}_\mathcal{E}$ of travel times is then as follows:

$$\begin{aligned} \bar{\mathbf{t}}_\mathcal{E} = \underset{\mathbf{t}_\mathcal{E}}{\text{argmin}} \quad & \|\mathcal{R}\mathbf{t}_\mathcal{E}\|_1, \\ \text{subject to} \quad & \mathcal{W},\ \mathbf{t}_{\mathcal{E},min} \leq \bar{\mathbf{t}}_\mathcal{E} \leq \mathbf{t}_{\mathcal{E},max}. \end{aligned} \tag{6}$$

4.2 Iterative Estimation

We refine $\bar{\mathbf{t}}_\mathcal{E}$ with an iterative process that alternates between *map-matching* and *travel-time estimation*. This design is based on the observation that many works take a sequential perspective [Hunter 2014; Kong et al. 2013; Li et al. 2017; Zhang et al. 2013]. As a consequence, the bias generated during *map-matching* (especially under the shortest-distance criterion) will get cascaded into *travel-time estimation* and affect the overall estimation accuracy. By resorting to an iterative process, as newly estimated travel times get more accurate, so does the map-matching, and vice-versa.

4.2.1 Map Matching. We adopt a simple strategy to compute the "true" path $\bar{k}_{s_1,s_2}$ between a pair of successive GPS data points $(s_1, s_2) \in \mathcal{S}_p$. Assuming Q_1 and Q_2 to denote the sets of candidate positions on the map for s_1 and s_2, respectively. This step returns $\mathcal{N} = \{\bar{k}_{s_1,s_2}\}_{\forall (s_1,s_2)\in\mathcal{S}_p}$, where:

$$\begin{aligned} \bar{k}_{s_1,s_2} &= \underset{\bar{k}\in\bar{\mathcal{K}}_{s_1,s_2}}{\text{argmin}} \|t(s_1,s_2) - \bar{t}(\bar{k})\|, \\ \text{with } \bar{\mathcal{K}}_{s_1,s_2} &= \left\{ \underset{k\in\mathcal{K}_{q_1,q_2}}{\text{argmin}}\ \bar{t}(k) \right\}_{\forall (q_1,q_2)\in Q_1\times Q_2}. \end{aligned} \tag{7}$$

4.2.2 Travel Time Estimation. Given the paths $\mathcal{N}$ assigned to each pair of successive GPS data points by the map-matching step, we now need to estimate a more accurate set of travel times $\bar{\mathbf{t}}_\mathcal{E}$ on individual road segments.

We model the travel times on road segments to each follow a probability distribution $\bar{t}(e) \sim \pi_e, \forall e \in \mathcal{E}$, where the distributions are parameterized by $\boldsymbol{\theta} = \{\theta_e\}_{e\in\mathcal{E}}$. Thus, we proceed to learn $\boldsymbol{\theta}$ through *maximum likelihood estimation* (MLE):

$$\underset{\boldsymbol{\theta}}{\text{maximize}}\ \mathcal{L}(\boldsymbol{\theta}|\mathcal{N}) = \sum_{\forall (s_1,s_2)\in\mathcal{S}_p} \log \pi(t(s_1,s_2)|\bar{k}_{s_1,s_2};\boldsymbol{\theta}), \tag{8}$$

where $\mathcal{L}$ is the likelihood function. Following the methodology from [Hofleitner et al. 2012], we assume that (1) the travel time on a road segment can be modeled by a univariate distribution, and (2) the travel-time distributions on road segments are pairwise independent. With these assumptions, we solve Equation 8 using the *expectation maximization* (EM) algorithm, for which the maximization problem from the M-step is for each road segment as follows:

$$\underset{\theta_e}{\text{maximize}} \sum_{\omega\in\Omega} w_\omega(e) \log \pi_e(t_\omega(e);\theta_e), \tag{9}$$

where π_e, θ_e, Ω, t_ω, and w_ω are as follows:

- π_e, θ_e: As in [Hunter 2014], we use the Gamma distribution Γ as π, because of its positive domain and long-tail-observation robustness properties. Thus, we have $\theta_e = (\alpha_e, \beta_e)$ where α_e is the shape and β_e is the scale. These quantities are also directly linked to the mean and standard deviation.

- Ω: As a road segment can be part of paths that link many different pairs of successive GPS data points, samples for each of these pairs must be included in Equation 9. Additionally, for more accurate estimations, we compute 100 samples for each pair. Thus, $\Omega = \{(s_1, s_2, i)\}_{\forall (s_1,s_2)\in\mathcal{S}_p, \forall i\in[\![1,100]\!]}$.

- t_ω: With the previous choice for π_e, θ_e and Ω, t_ω is computed as follows [Hunter 2014]:

$$\begin{aligned} \forall \omega \in \Omega,\ \forall e \in \bar{k}_{s_1,s_2},\ \ t_\omega(e) &= t(s_1,s_2)\frac{A_\omega(e)}{\sum_{e\in\mathcal{E}} A_\omega(e)} \\ \text{with } A_\omega(e) &\sim \Gamma(\alpha_e, \frac{\beta_e}{t(s_1,s_2)}), \end{aligned} \tag{10}$$

- w_ω: A weight $w_\omega(e)$ is then computed as the distance between $t_\omega(e)$ and $A_\omega(e)$.

After solving Equation 9, a new estimation of $\bar{\mathbf{t}}_\mathcal{E}$ is computed, where for each road segment $e \in \mathcal{E}$, $\bar{t}(e)$ is computed as the mean of π_e. From here, the iterative estimation loops back to the *map-matching* step. The entire process stops after 10 iterations (determined empirically), with the refined $\bar{\mathbf{t}}_\mathcal{E}$ as output.

4.3 Bilevel Optimization

The previous steps allowed us to have the estimated travel times $\bar{\mathbf{t}}_\mathcal{E}$ on road segments with GPS data coverage. Then the objective of this step is to compute the flows on all road segments (including those without data coverage).

First, we convert $\bar{\mathbf{t}}_\mathcal{E} = [\bar{t}(e)]^\intercal_{e\in\mathcal{E}}$ to $\bar{\mathbf{f}}_\mathcal{E} = [\bar{f}(e)]^\intercal_{e\in\mathcal{E}}$ using the relation from Appendix A.1. Then, we derive the target OD pairs $\bar{\mathbf{u}}$ based on the ratio of estimated flows and loop-detector measurements on the same road segment [Yang et al. 2017]. By having $\bar{\mathbf{f}}_\mathcal{E}$ from the previous steps, we can compute flows on all road segments $\hat{\mathbf{f}}_\mathcal{E}$ and the corresponding flows between OD pairs $\hat{\mathbf{u}}$ through the following minimization [Cascetta and Nguyen 1988]:

[2]We set the threshold to be 120% due to two reasons: 1) it regulated the range of traffic flows; 2) by examining map-matched GPS data, traffic flow velocity that exceeds 120% is rare.

$$\begin{aligned} \underset{\hat{\mathbf{u}}}{\text{minimize}} \quad & \mathcal{F}_1(\hat{\mathbf{u}}, \bar{\mathbf{u}}) + \mathcal{F}_2(\hat{\mathbf{f}}_{\mathcal{E}}, \bar{\mathbf{f}}_{\mathcal{E}}) \\ \text{subject to} \quad & \hat{\mathbf{f}}_{\mathcal{E}} = \mathcal{M}(\hat{\mathbf{u}}),\ \hat{\mathbf{u}} \geq 0, \end{aligned} \tag{11}$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are generalized distance functions. $\mathcal{M}$ is the *assignment map* which determines $\mathbf{P}_{\mathcal{E}}$. If $\mathcal{M}$ follows Wardrop's principle [Wardrop 1952], then Equation 11 becomes a bilevel optimization problem [Yang et al. 1992]: the upper level minimizes the distances of estimated OD pairs and traffic flows to their corresponding measurements, while the lower level satisfies the *user equilibrium*. For $\mathcal{F}_1$ and $\mathcal{F}_2$, we select the generalized least squares (GLS) estimator [Bera and Rao 2011], as it permits different weighting schemes of $\bar{\mathbf{u}}$ and $\bar{\mathbf{f}}_{\mathcal{E}}$. We further assume that $\bar{\mathbf{u}}$ and $\bar{\mathbf{f}}_{\mathcal{E}}$ are results from the following stochastic system of equations:

$$\bar{\mathbf{u}} = \hat{\mathbf{u}} + \epsilon_1,\ \bar{\mathbf{f}}_{\mathcal{E}} = \hat{\mathbf{f}}_{\mathcal{E}} + \epsilon_2. \tag{12}$$

Then, Equation 11 can be explicitly written as:

$$\begin{aligned} \underset{\hat{\mathbf{u}}}{\text{minimize}} \quad & \eta(\bar{\mathbf{u}} - \hat{\mathbf{u}})^\intercal U^{-1}(\bar{\mathbf{u}} - \hat{\mathbf{u}}) \\ & + (1-\eta)(\bar{\mathbf{f}}_{\mathcal{E}} - \hat{\mathbf{f}}_{\mathcal{E}})^\intercal V^{-1}(\bar{\mathbf{f}}_{\mathcal{E}} - \hat{\mathbf{f}}_{\mathcal{E}}), \\ \text{subject to} \quad & \hat{\mathbf{f}}_{\mathcal{E}} = \mathcal{M}(\hat{\mathbf{u}}),\ \hat{\mathbf{u}} \geq 0,\ \hat{\mathbf{f}}_{\mathcal{E}} \geq 0,\ 0 \leq \eta \leq 1, \end{aligned} \tag{13}$$

where U and V are variance-covariance matrices of ϵ_1 and ϵ_2, respectively; $\mathbb{E}(\epsilon_1) = 0$ and $\mathbb{E}(\epsilon_2) = 0$ are derived from experiments in [Cascetta and Nguyen 1988]; $\eta \in [0,1]$ is the weighting factor. When $\eta = 1$, $\bar{\mathbf{f}}_{\mathcal{E}}$ is ignored, and the estimation is purely based on $\bar{\mathbf{u}}$; when $\eta = 0$, the estimation is solely based on $\bar{\mathbf{f}}_{\mathcal{E}}$. Equation 13 can be solved iteratively: in the upper level we solve the GLS estimator using quadratic programming; in the lower level we approximate *user equilibrium* using the *one-shot* function within SUMO [Krajzewicz et al. 2012]. As a result of this procedure, we can use the newly computed $\hat{\mathbf{f}}_{\mathcal{E}}$ on road segments where GPS data is unavailable.

5 DYNAMIC DATA COMPLETION

In order to generate accurate dynamic traffic simulations, we use a microscopic simulation algorithm. However, to ensure that the simulation of any given area respects flows at the boundaries and previously reconstructed traffic, we allow the algorithm to alter the values of the turning ratios at intersections (where a car will turn when arriving at an intersection). We choose the turning ratios as optimization variables because: 1) turning ratios implicitly encode several aspects, e.g. traffic light logic, 2) driving behaviors, such as lane changing, are limited at intersections, and 3) detailed road information (e.g. intersection logic) is difficult to obtain. Consequently, in most microscopic traffic simulators, specifying turning ratios is one of the main mechanisms to start a simulation. Furthermore, the split flows can be used to form a metamodel and compute estimation errors from the previous reconstructed results. Lastly, our algorithm explained in this section can cope with multivariable optimization if information of other simulation parameters is provided.

We denote the vector of all turning ratios by $\mathbf{x} = [x_{v,e}]_{\forall e \in v, \forall v \in \mathcal{V}}$, and name it as a *decision point*. In order to systematically derive a decision point $\mathbf{x}^*$ in a simulation region that not only meets OD demands but also agrees with the previous estimated traffic conditions, we rely on the following optimization task:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\text{argmin}} \quad \bar{F}(\mathbf{x}; \rho) \equiv \mathbb{E}[F(\mathbf{x}; \rho)], \tag{14}$$

where $\bar{F}$ is the objective function, and F is a stochastic network performance measure. The distribution of F depends on the decision point $\mathbf{x}$ and exogenous parameters ρ, which record a network topology and road-segment metrics. Every simulation run with $\mathbf{x}$ is a realization of F, which involves sampling many other distributions that account for the stochastic nature of traffic (e.g. driver differences). Assume we run r independent simulations with a given $\mathbf{x}$, the objective function then can be approximated as:

$$\hat{F}(\mathbf{x}; \rho) = \frac{1}{r} \sum_{i=1}^{r} F_i(\mathbf{x}; \rho). \tag{15}$$

However, using a simulation algorithm in an optimization context can be very costly. This motivates us to adopt the metamodel-based simulation optimization approach.

5.1 Metamodel-Based Simulation Optimization

A metamodel simplifies simulation optimization, as it is typically a deterministic function rather than a stochastic simulation, with much lower computational complexity than the objective function. Therefore, one way to circumvent the issues of using a microscopic traffic simulator in a simulation-optimization loop is to develop a deterministic metamodel to replace the stochastic simulation response. The metamodel is usually less realistic in terms of the modeling capability, but much cheaper to evaluate.

The most common metamodels are *functional* metamodels, which are general-purpose functions and can be used to approximate arbitrary objective functions. Often, they are results of a linear combination of basis functions such as low-order polynomials, spline models, and radial basis functions [Conn et al. 2009]. However, they require large numbers of decision points to be fitted, since they do not take the structure of an underlying problem into account. This means that we need to run the simulator on many decision points in order to fit a well-performing metamodel. This procedure is expensive and to a certain degree defeats the purpose of using a metamodel. Therefore, we use a metamodel that contains not only a *functional* component but also a *physical* component which encodes the underlying problem to achieve high efficiency.

We build this physical component based on the classical *flow conversion equation*, also known as the *traffic equation* [Osorio 2010]:

$$f(e_1) = \gamma(e_1) + \sum_{e_2 \in C} p(e_1, e_2) f(e_2),\ \forall e_1 \in C, \tag{16}$$

where C represents the set of road segments in the simulation region, $\gamma(e_1)$ is the external flow injected into road segment e, and $p(e_1, e_2)$ is the transition probability from road segment e_1 to road segment e_2. The exogenous parameters in Equation 16 are external flows and transition probabilities. By having them, we can execute both the traffic equation and the traffic simulator to obtain flow measurements on all road segments in a simulation region. Denoting the subset of road segments with already estimated traffic conditions from GPS data as $\mathcal{A} \in C$, the estimated flows as $\mathbf{f}^E = \{f(e)^E\}_{e \in \mathcal{A}}$, the propagated flows using the traffic equation

as $\mathbf{f}^T = \{f(e)^T\}_{e\in\mathcal{A}}$, and the simulated flows using a traffic simulator as $\mathbf{f}^S = \{f(e)^S\}_{e\in\mathcal{A}}$, we define the estimate of the objective function derived from the simulator with one simulation run ($r = 1$ in Equation 15) as:

$$\hat{F}(\mathbf{x};\rho) = F(\mathbf{x};\rho) = \|\mathbf{f}^S - \mathbf{f}^E\|_2. \tag{17}$$

We define the approximation of the objective function derived from the traffic equation as:

$$T(\mathbf{x};\rho) = \|\mathbf{f}^T - \mathbf{f}^E\|_2. \tag{18}$$

The metamodel is then constructed as a combination of the physical component T and a functional component Φ:

$$M(\mathbf{x};\alpha,\beta,\rho) = \alpha T(\mathbf{x};\rho) + \Phi(\mathbf{x};\boldsymbol{\beta}), \tag{19}$$

where α (initially set to 0.5) and $\boldsymbol{\beta}$ (initially set to $\mathbf{1}$) are parameters of the metamodel. The functional component, Φ, is chosen to be a quadratic polynomial [Osorio and Bierlaire 2013]:

$$\Phi(\mathbf{x};\boldsymbol{\beta}) = \beta_1 + \sum_{i=1}^{|\mathbf{x}|}\beta_{i+1}x_i + \sum_{i=1}^{|\mathbf{x}|}\beta_{i+|\mathbf{x}|+1}x_i^2, \tag{20}$$

where $|\mathbf{x}|$ is the dimension of $\mathbf{x}$; x_i and β_i are the ith elements of $\mathbf{x}$ and $\boldsymbol{\beta}$ respectively. The quadratic polynomial provides Taylor-type bounds, serves as a general term within a metamodel formulation, and ensures global convergences [Conn et al. 2009]. In order to fit the metamodel, we rely on the decision points that have been evaluated using both the traffic equation and the traffic simulator. Denoting $\mathcal{X}$ as the pool of the decision points, we can fit the metamodel (i.e. compute α and $\boldsymbol{\beta}$) by solving:

$$\begin{aligned}\underset{\alpha,\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i=1}^{|\mathcal{X}|}\left(w_i\left(\hat{F}(\mathbf{x}_i;\rho) - M(\mathbf{x}_i;\alpha,\boldsymbol{\beta},\rho)\right)\right)^2 \\ & + (w_0\cdot(\alpha-1))^2 + \sum_{j=1}^{2|\mathbf{x}|+1}(w_0\cdot\beta_j)^2,\end{aligned} \tag{21}$$

where w_0 is a fixed constant. w_i is the weight associating each $\mathbf{x}_i$ and a new decision point $\mathbf{x}_{new}$ at each iteration during the optimization, and is computed as $w_i = 1/(1 + \|\mathbf{x}_{new} - \mathbf{x}_i\|_2)$, representing the *inverse distance* [Atkeson et al. 1997]. The first term in Equation 21 represents the weighted distance between simulated results and estimated results. The remaining terms in Equation 21 guarantee the least square matrix to have a full rank.

5.2 Algorithmic Steps

Following the framework proposed in [Conn et al. 2009] and its adaptation in [Osorio and Bierlaire 2013], we combine the metamodel with the derivative-free trust-region algorithm to solve the optimization problem from Equation 14, which can now be expressed at any given iteration as:

$$\begin{aligned}\mathbf{x}_{new} = \underset{\mathbf{x}}{\text{argmin}} \quad & M(\mathbf{x};\alpha,\boldsymbol{\beta},\rho) = \alpha T(\mathbf{x};\rho) + \Phi(\mathbf{x};\boldsymbol{\beta}), \\ \text{subject to} \quad & \|\mathbf{x}^* - \mathbf{x}\|_2 \le \Delta,\ 0 \le x \le 1, \forall x \in \mathbf{x},\end{aligned} \tag{22}$$

where $\mathbf{x}^*$ is the best decision point so far and Δ is the current trust-region radius.

The specific algorithmic steps are as follows:

- **Step 1:** Initialize $\mathcal{X}$ as containing 5 randomly sampled decision points, evaluate each of them using both the traffic equation and **simulator**, arbitrarily set $\mathbf{x}^*$ as a any element of $\mathcal{X}$, and compute α and $\boldsymbol{\beta}$ (Equation 21).
- **Step 2:** Use Equation 22 to compute $\mathbf{x}_{new}$.
- **Step 3:** Compute $\hat{F}(\mathbf{x}_{new};\rho)$ (**simulator run**). Compute the relative improvement $\tau = \frac{\hat{F}(\mathbf{x}_{new}) - \hat{F}(\mathbf{x}^*)}{M_i(\mathbf{x}_{new}) - M_i(\mathbf{x}^*)}$. If $\tau \ge 1e-3$, **accept** $\mathbf{x}_{new}$ and set $\mathbf{x}^* := \mathbf{x}_{new}$, otherwise **reject** $\mathbf{x}_{new}$. In any case, add $\mathbf{x}_{new}$ to $\mathcal{X}$, and compute α and $\boldsymbol{\beta}$ (Equation 21).
- **Step 4:** If α and $\boldsymbol{\beta}$ have not changed much in **step 3**, i.e. $\frac{\|(\alpha_{new},\boldsymbol{\beta}_{new}) - (\alpha_{old},\boldsymbol{\beta}_{old})\|}{\|(\alpha_{old},\boldsymbol{\beta}_{old})\|} \le 0.1$, then add a new randomly sampled decision point to $\mathcal{X}$, evaluate it using both the traffic equation and **simulator**, and compute α and $\boldsymbol{\beta}$ (Equation 21).
- **Step 5:** Update the trust-region radius:

$$\Delta := \begin{cases} \min\{1.1\times\Delta, 100\} & \text{if } \tau > 1e-3, \\ \max\{0.9\times\Delta, 0.1\} & \text{if } \tau \le 1e-3 \text{ and} \\ & 5 \text{ consecutive rejections of } \mathbf{x}, \\ \Delta & \text{otherwise.}\end{cases} \tag{23}$$

- **Step 6:** Exit the loop when the maximum number of allowed simulator runs (20) is reached; otherwise go to **step 2**.

When the algorithm stops, $\mathbf{x}^*$ can then be used to generate a simulation, thus ending dynamic data completion.

In practice, instead of running this procedure on the whole network, we adopt a *decomposition approach* and separate the network into sub-networks which are modeled independently. For these small sub-networks, we consider nodes with no predecessors as *artificial origins* and nodes without successors as *artificial destinations* (i.e. locations where vehicles respectively enter and exit the sub-network). Additionally, observing that vehicles rarely navigate in loops, we extract directed acyclic graphs (DAG) from these sub-networks. In order to form a decision point, we consider the turning ratios of nodes at the start of DAG edges that do not have accurate flow estimations (and treat the rest of the turning ratios as constants).

6 RESULTS

In this section, we first present evaluations of *Initial Traffic Reconstruction* and *Dynamic Data Completion*. Then, we demonstrate the visualization and animation results using our method[3].

6.1 Initial Traffic Reconstruction

We evaluate our approach using a real road network as the benchmark, a set of heuristic network travel times inferred from a real GPS dataset, and a set of synthetic GPS traces generated as the ground truth to test the performance.

We evaluate our method using the road network from downtown San Francisco. For our experiments, we have established synthetic datasets with GPS traces and precise network travel times. We refer to a complete set of travel times on the road network as a traffic condition. In total, we have generated 10 traffic conditions via solving the system optimal (SO) model, and 24 traffic conditions via the Timestamp model. For each condition, we have sampled 315,000

[3]The simulation resource can be found at http://gamma.cs.unc.edu/CityFlowRecon.

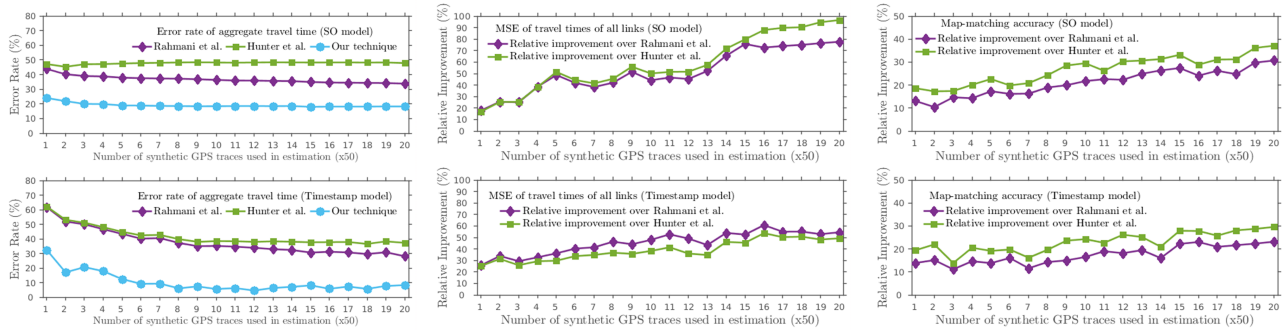

Fig. 3. From left to right, the top diagrams show results generated using network travel times using the system optimal (SO) model, while the bottom diagrams show corresponding results of network travel times using the Timestamp model. Left: The relative error rates (%) of various methods of aggregated travel time across the road network. Center: The relative improvements (%) of travel times of all road segments measured in MSE. Right: The relative improvements (%) of map-matching accuracy measured using successful identification rates of road segments. In summary, our technique achieves consistently more accurate results over the state of the art, as the number of GPS traces used in recovering network travel times increases.

GPS traces for evaluation. We refer interested readers to Appendix A.2 for details on generating this synthetic test data.

We compare our technique with two state-of-the-art methods, namely Hunter et al. [2014] and Rahmani et al. [2015]. The results are shown in Figure 3. We evaluate our technique using three metrics. The first metric is the error rate of the aggregate travel time across the entire road network. The minimum error rate of our technique is 18% (top left of Figure 3) and 8% (bottom left of Figure 3), while the other two approaches have much higher error rates. The second metric is the relative improvement of our technique over existing methods on travel times on all road segments. The maximum relative improvements over Hunter et al. [2014] and Rahmani et al. [2015] under the SO model (top center of Figure 3) are 78% and 97%, and under the Timestamp model (bottom center of Figure 3) are 54% and 49%. The third metric evaluates the map-matching accuracy. The maximum relative improvements of our method over Hunter et al. [2014] and Rahmani et al. [2015] are 28% and 34% under the SO model, and 19% and 25% under the Timestamp model. These results are shown on the right of Figure 3. The details of these experiments and the evaluation of the bilevel optimization can be found in Appendix A.2.

6.2 Metamodel-Based Simulation Optimization

We compare our technique against the approach that only uses the traffic simulator (i.e. SUMO [Krajzewicz et al. 2012]) on various OD demands and road networks.

We set up our experiments by starting at the center of the road network in San Francisco and gradually increase the radius to retrieve networks with 20 to 5,000 road segments. This step results in 48 road networks. For all DAGs constructed in all networks, the OD demand varies from $1{,}000 veh/hr$ to $10{,}000 veh/hr$ with an increment of $1{,}000 veh/hr$.

In Figure 4, we show the accuracy of our technique compared to the approach using only the traffic simulator. For each road network in the experiment, we randomly select an intersection and randomly assign the turning ratios at this intersection. The assigned turning ratios (i.e. a decision point) are treated as the ground truth for the two approaches to recover. Since the most accurate method for recovering a traffic condition is the microscopic simulator [Sewall et al. 2011b], we use the error level of the microscopic simulator to the ground truth as the baseline. For each decision point, we first compute the difference between the recovered value using our technique to the ground truth, as well as the difference between the recovered value by the simulator to the ground truth. Then, we subtract these two error differences and represent the result as a gray cross in Figure 4. The mean, minimum, and maximum values of the average deviations to the simulator (indicated by the solid line in Figure 4), are respectively 7.8%, 0%, and 13%. In many cases, our technique even outperforms the simulation with much smaller differences to the ground truth, as indicated by the negative values in Figure 4. 3

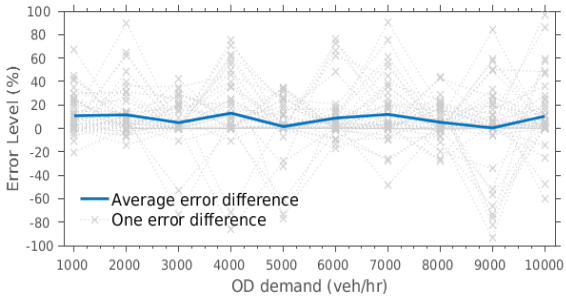

Fig. 4. **The error level of ours vs. simulation-only approach:** For a given road network and a specific OD demand, we first compute the differences between the two methods with respect to the ground truth. Then, we subtract these two differences to obtain one error difference measure (i.e. a gray cross). The mean, minimum, and maximum values of the average error level (indicated by the solid line) are respectively 7.8%, 0%, and 13%. In many cases, our technique even outperforms the simulation with much smaller differences to the ground truth, as indicated by the negative values.

In the second study, we analyze the performance speedup achieved by our technique over the simulation-only approach. The results are shown in Figure 5. Each gray cross indicates a speedup measure for

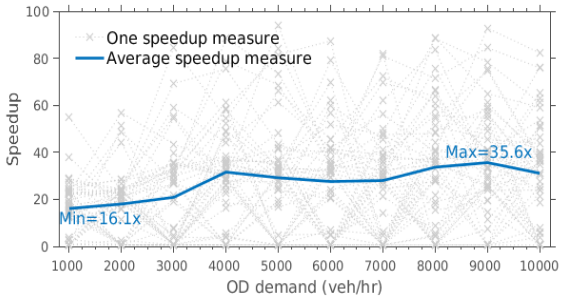

Fig. 5. **The performance speedup of our technique over the simulation-only approach:** Our technique in *on average* about $27.2x$ faster, with maximum and minimum performance gains of $35.6x$ and $16.1x$. The maximum observed performance gain of a single speedup measure is *over 90x* (at OD demand = 9,000).

a particular road network and a specific OD demand. The highest speedup is over $90x$. On average, the maximum speedup is about $35.6x$ and the minimum speedup is about $16.1x$. There also exist several cases where our technique demonstrates rather negligible speedups (i.e. values close to 0). This is because, for these cases, the initial random guess of the decision point is close to the ground truth. Consequently, either method can achieve a quick convergence. In reality, we expect these scenarios to happen rarely. Finally, in nearly all experiments, our technique converges in the first five iterations.

6.3 Traffic Visualization and Animation

To demonstrate the effectiveness of our approach, we use GPS data from taxicabs in San Francisco to perform traffic visualization and animation on a metropolitan scale. This dataset from [Piorkowski et al. 2009] consists of GPS traces from 536 taxicabs. The overall number of GPS traces is *over 11 million*, resulting in *over 4.7 million kilometers* of traveled distance.

In Figure 6 (top row), we show the visualization results over four time intervals: Sunday 9AM, Tuesday 9AM, Thursday Noon, and Friday 7PM, which exemplify *weekend morning traffic*, *weekday morning traffic*, *weekday mid-day traffic*, and *weekday evening traffic*, respectively. In San Francisco, first the congestion level of Sunday 9AM is low compared to the rest of the time intervals across all areas. Second, the congestion of Tuesday 9AM is more severe in the north-west, central-west, and central-east areas (residential regions) than the same areas in other time intervals. Lastly, the north-east region (downtown commercial and financial districts) of Tuesday 9AM and Friday 7PM appear to have more severe congestion than other time intervals.

In addition to the qualitative results, we have quantitatively compared our reconstructed results with loop-detector data[4] extracted from the same location and time periods as the GPS data. The loop-detector data represents relatively accurate measurements and is often regarded as the de-facto standard for evaluating GPS-based estimations [Work et al. 2010]. After applying a filtering process, which is explained in Appendix A.3, our reconstruction approximates the accurate loop-detector readings with small losses (around $1m/s$) in the speed measurement. This validation result can be found in the bottom row of Figure 6.

As a result of the dynamic data completion, we obtain turning ratios that not only can lead to simulations that respect the reconstructed conditions, but also serve as a means to complete missing information in regions where the GPS data is scarce. The results can be illustrated using both 2D and 3D traffic animation for various virtual-world applications. In Figure 7, we provide 2D traffic animation of four regions in San Francisco[5] using the reconstructed traffic on Friday at 7PM. This 2D animation can be used to study dynamic traffic patterns at a metropolitan scale. The downtown area is further modeled in a 3D *Virtual San Francisco* to showcase the potential of embedding real-world traffic in a virtual world for immersive VR experiences and virtual tourism applications (see Figure 8). We have used the real-world GPS data throughout our experiments. Though such data may not be representative of overall traffic patterns, our approach is independent from data sources. The effectiveness is demonstrated using both synthetic datasets (Figure 3) and real-world datasets (Figure 6).

7 CONCLUSION AND FUTURE WORK

We have presented an efficient algorithm to reconstruct city-scale traffic from GPS data using statistical learning. To address the issues with incomplete and/or sparse data, a metamodel-based simulation optimization is proposed to dynamically bridge the "gap" between the reconstructed traffic learned from GPS data and the simulated traffic where the data is incomplete or missing. This approach is able to perform visualization of city-scale traffic, as well as data-driven 2D and 3D traffic animation in a virtual environment. With more GPS datasets being made available and released to public, e.g. Mobile Century [Herrera et al. 2010], T-Drive [Microsoft 2010], GeoLife [Microsoft 2009], and Uber Movement [Uber 2017], we believe future research would be abundant.

Although the proposed approach is specialized for traffic reconstruction, similar approaches can be developed to reconstruct aggregate dynamics of other multi-agent systems using spatial-temporal data, such as schools of fishes, flocks of birds, and swarms of insects. More importantly, the idea of learning from mobile sensor data as well as the concept of using metamodel-based optimization to refine the simulation parameters and to accelerate local fitting for large-scale motion reconstruction are generalizable to many other domains in addition to computer graphics.

Our framework has some limitations. First, the accompanying animation in the supplementary video may exhibit some odd behaviors of individual vehicles. This is implementation specific due to the use of SUMO that decides lane priorities at intersections (different from turning ratios) and stochastically sample vehicle behaviors, such as acceleration, deceleration, and lane-changing frequency. However, such occasional visual artifacts do not affect our results macroscopically and our algorithm is independent of a specific traffic simulator. Other microscopic simulators and more advanced vehicle kinematic models can be easily integrated with

[4]The loop-detector data is obtained from http://pems.dot.ca.gov/.

[5]The number of lanes of a road segment is decided by the digital map.

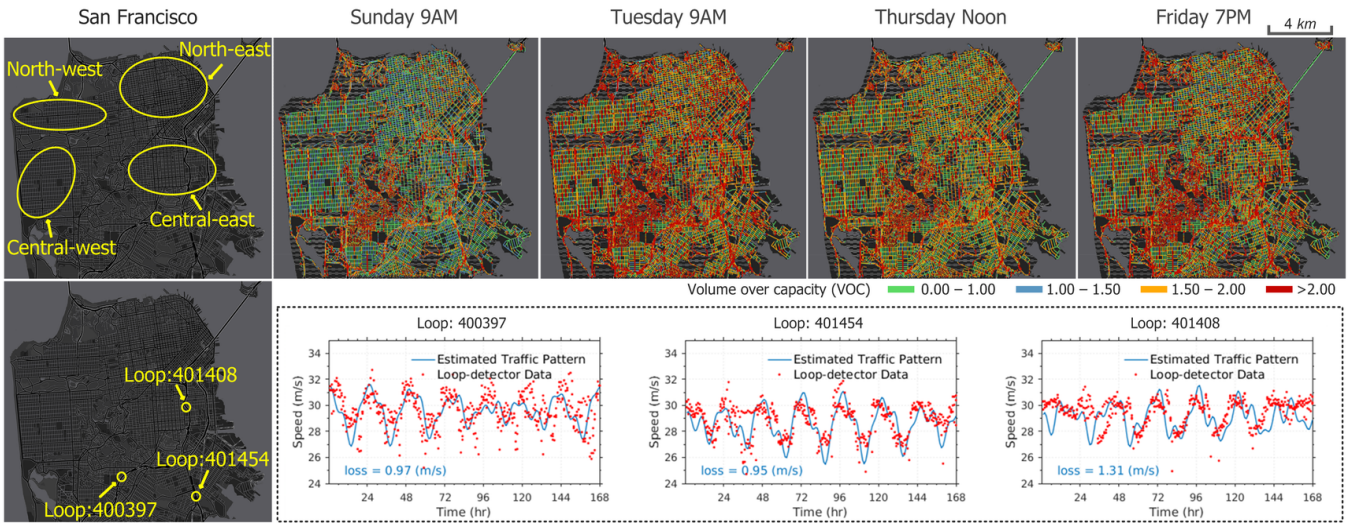

Fig. 6. **Qualitative (visualization) and quantitative analysis of traffic in San Francisco.** Top: We use four time periods in a week, namely Sunday 9AM, Tuesday 9AM, Thursday Noon, and Friday 7PM, to illustrate weekend vs. weekday and morning vs. evening traffic. The traffic is measured by *volume over capacity* (VOC) which is computed as $\sum_{e \in \mathcal{E}} (f(e)/c(e))$ where $c(e)$ is the capacity of the road segment e defined in Appendix A.1. Bottom: We compare our reconstructed results (after a filtering process explained in Appendix A.3) with the data of three loop detectors. The results show small losses (around $1m/s$) in the speed measurement.

Fig. 7. **2D traffic animation of regions in San Francisco:** Northeast (top left), Central-East (top center), Central (top Right), Northwest (bottom). We have exaggerated the headlights and adopted an evening time period (i.e. Friday 7PM) to make vehicles more visible.

our framework to further improve the animation quality. Second, the accuracy of our reconstruction is limited by the available data. While we can maintain the accuracy down to the road-segment level, high frequency vehicle positions can not be precisely modeled, as such information is largely missing from the data. Third, while we can satisfy the flow conservation in each sub-network that runs our technique, such a relationship is difficult to maintain at a city scale. As we expect more in-road sensors to be deployed in the future, this problem will be alleviated.

There are a number of future directions. Algorithmically, one extension is to combine other data sources (in-road sensors, video streams, or surveying) with GPS traces to further improve the reconstruction accuracy. Another possibility is to incorporate a macroscopic traffic simulator, so that we can dynamically switch between simulators of varying fidelities to further reduce the computational cost. Application-wise, a virtual tourism system, a route planning [Wilkie et al. 2011] and navigation system [Wilkie et al. 2014], or an autonomous vehicle training system can immediately

Fig. 8. **3D traffic animation:** a perspective overview (left), a topdown view (center), and a driver's view (right).

benefit from this work by incorporating and visualizing traffic conditions reflecting real-world dynamics.

ACKNOWLEDGEMENT

The authors would like to thank US Army Research Office and UNC Arts & Science Foundation for their support.

REFERENCES

Natalia Andrienko and Gennady Andrienko. 2006. *Exploratory analysis of spatial and temporal data: a systematic approach.*

Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. 1997. Locally Weighted Learning. *Artif. Intell. Rev.* 11, 1-5 (1997), 11–73.

Salvador Bayarri, Marcos Fernandez, and Mariano Perez. 1996. Virtual Reality for Driving Simulation. *Commun. ACM* 39, 5 (1996), 72–76.

Sharminda Bera and K. V. Krishna Rao. 2011. Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport\Trasporti Europei n.* (2011), 3–23.

Huikun Bi, Tianlu Mao, Zhaoqi Wang, and Zhigang Deng. 2016. A data-driven model for lane-changing in traffic simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 149–158.

Ennio Cascetta and Sang Nguyen. 1988. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B: Methodological* 22, 6 (1988), 437–455.

Pablo Samuel Castro, Daqing Zhang, and Shijian Li. 2012. Urban traffic modelling and prediction using large scale taxi GPS traces. In *International Conference on Pervasive Computing*. 57–72.

Qianwen Chao, Zhigang Deng, Jiaping Ren, Qianqian Ye, and Xiaogang Jin. 2017. Realistic Data-Driven Traffic Flow Animation Using Texture Synthesis. *IEEE Transactions on Visualization and Computer Graphics* (2017).

Qianwen Chao, Jingjing Shen, and Xiaogang Jin. 2013. Video-based personalized traffic learning. *Graphical Models* 75, 6 (2013), 305–317.

Panayiotis Charalambous and Yiorgos Chrysanthou. 2014. The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation. *Computer Graphics Forum* 33, 8 (2014), 95–108.

Andrew Conn, Katya Scheinberg, and Luis N. Vicente. 2009. *Introduction to derivative-free optimization.*

Nivan Ferreira, Jorge Poco, Huy T Vo, Juliana Freire, and Cláudio T Silva. 2013. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2149–2158.

Ignacio Garcia-Dorado, Daniel G. Aliaga, Saiprasanth Bhalachandran, Paul Schmid, and Dev Niyogi. 2017. Fast Weather Simulation for Inverse Procedural Design of 3D Urban Models. *ACM Trans. Graph.* 36, 2 (2017), 21:1–21:19.

I. Garcia-Dorado, D. G. Aliaga, and S. V. Ukkusuri. 2014. Designing large-scale interactive traffic animations for urban modeling. *Computer Graphics Forum* 33, 2 (2014), 411–420.

E. Hato, M. Taniguchi, Y. Sugie, M. Kuwahara, and H. Morita. 1999. Incorporating an information acquisition process into a route choice model with multiple information sources. *Transportation Research Part C: Emerging Technologies* 7, 2 (1999), 109–129.

Bruce Hellinga, Pedram Izadpanah, Hiroyuki Takada, and Liping Fu. 2008. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. *Transportation Research Part C: Emerging Technologies* 16, 6 (2008), 768–782.

Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang (Jeff) Ban, Quinn Jacobson, and Alexandre M. Bayen. 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies* 18, 4 (2010), 568–583.

Ryan Herring, Aude Hofleitner, Pieter Abbeel, and Alexandre Bayen. 2010. Estimating arterial traffic conditions using sparse probe data. In *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on.* 929–936.

Aude Hofleitner, Ryan Herring, and Alexandre Bayen. 2012. Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach. In *91st Transportation Research Board Annual Meeting.*

Timothy Hunter. 2014. Large-Scale, Low-Latency State Estimation Of Cyberphysical Systems With An Application To Traffic Estimation. *Ph.D. Thesis* (2014).

Timothy Hunter, Pieter Abbeel, and Alexandre Bayen. 2014. The path inference filter: model-based low-latency map matching of probe vehicle data. *Intelligent Transportation Systems, IEEE Transactions on* 15, 2 (2014), 507–529.

Eunjung Ju, Myung Geol Choi, Minji Park, Jehee Lee, Kang Hoon Lee, and Shigeo Takahashi. 2010. Morphable Crowds. *ACM Trans. Graph.* 29, 6 (2010), 140:1–140:10.

Pushkin Kachroo and Shankar Sastry. 2016. Travel time dynamics for intelligent transportation systems: theory and applications. *IEEE Transactions on Intelligent Transportation Systems* 17, 2 (2016), 385–394.

Qing-Jie Kong, Qiankun Zhao, Chao Wei, and Yuncai Liu. 2013. Efficient traffic state estimation for large-scale urban road networks. *IEEE Transactions on Intelligent Transportation Systems* 14, 1 (2013), 398–407.

Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. 2012. Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements* 5, 3-4 (2012), 128–138.

Jon Kuhl, Douglas Evans, Yiannis Papelis, Richard Romano, and Ginger Watson. 1995. The Iowa Driving Simulator: An Immersive Research Environment. *Computer* 28, 7 (1995), 35–41.

Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. 2007. Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 109–118.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by Example. *Computer Graphics Forum* 26, 3 (2007), 655–664.

Li Li, Xiqun Chen, and Lei Zhang. 2014. Multimodel ensemble for freeway traffic state estimations. *IEEE Transactions on Intelligent Transportation Systems* 15, 3 (2014), 1323–1336.

Minglei Li, Peter Wonka, and Liangliang Nan. 2016. Manhattan-World Urban Reconstruction from Point Clouds. In *European Conference on Computer Vision*. 54–69.

Weizi Li, Dong Nie, David Wilkie, and Ming C. Lin. 2017. Citywide Estimation of Traffic Dynamics Via Sparse GPS Traces. *IEEE Intelligent Transportation Systems Magazine* 9, 3 (2017), 100–113.

Wen-Chieh Lin, Sai-Keung Wong, Cheng-Hsing Li, and Richard Tseng. 2016. Generating Believable Mixed-Traffic Animation. *IEEE Transactions on Intelligent Transportation Systems* 17, 11 (2016), 3171–3183.

Xuequan Lu, Zonghui Wang, Mingliang Xu, Wenzhi Chen, and Zhigang Deng. 2014. A personality model for animating heterogeneous traffic behaviors. *Computer animation and virtual worlds* 25, 3-4 (2014), 361–371.

Tianlu Mao, Hua Wang, Zhigang Deng, and Zhaoqi Wang. 2015. An efficient lane model for complex traffic simulation. *Computer Animation and Virtual Worlds* 26, 3-4 (2015), 397–403.

Microsoft. 2009. GeoLife Project. www.microsoft.com/en-us/research/project/geolife-building-social-networks-using-human-location-history/. (2009).

Microsoft. 2010. T-Drive Project. www.microsoft.com/en-us/research/project/t-drive-driving-directions-based-on-taxi-traces/. (2010).

MIT. 2011. MIT Intelligent Transportation Systems. its.mit.edu/. (2011).

P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer. 2013. A Survey of Urban Reconstruction. *Computer Graphics Forum* 32, 6 (2013), 146–177.

Carolina Osorio. 2010. *Mitigating network congestion: analytical models, optimization methods and their applications.* Ph.D. Dissertation.

Carolina Osorio and Michel Bierlaire. 2013. A simulation-based optimization framework for urban transportation problems. *Operations Research* 61, 6 (2013), 1333–1345.

Carolina Osorio, Gunnar Flötteröd, and Chao Zhang. 2015. A metamodel simulation-based optimization approach for the efficient calibration of stochastic traffic simulators. *Transportation Research Procedia* 6 (2015), 213–223.

Mikko Perttunen, Vassilis Kostakos, Jukka Riekki, and Timo Ojala. 2015. Urban traffic analysis through multi-modal sensing. *Personal and Ubiquitous Computing* 19, 3-4 (2015), 709–721.

Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.org/epfl/mobility/20090224. (2009).

Mohammed Quddus and Simon Washington. 2015. Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies* 55 (2015), 328–339.

Mahmood Rahmani, Erik Jenelius, and Haris Koutsopoulos. 2015. Non-parametric estimation of route travel time distributions from low-frequency floating car data. *Transportation Research Part C: Emerging Technologies* 58 (2015), 343–362.

David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015. 2015 Urban Mobility Scorecard. *Texas A&M Transportation Institute and INRIX* (2015).

Jason Sewall, Jur Van Den Berg, Ming Lin, and Dinesh Manocha. 2011a. Virtualized traffic: Reconstructing traffic flows from discrete spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics* 17, 1 (2011), 26–37.

Jason Sewall, David Wilkie, and Ming C. Lin. 2011b. Interactive Hybrid Simulation of Large-scale Traffic. *ACM Trans. Graph.* 30, 6 (2011), 135:1–135:12.

Jason Sewall, David Wilkie, Paul. Merrell, and Ming C. Lin. 2010. Continuum Traffic Simulation. *Computer Graphics Forum* 29, 2 (2010), 439–448.

Yosef Sheffi. 1985. *Urban Transportation Network*. Pretince Hall.

Jingjing Shen and Xiaogang Jin. 2012. Detailed traffic animation for urban road networks. *Graphical Models* 74, 5 (2012), 265–282.

TAC. 2017. TAC's virtual reality driving school. www.smh.com.au/victoria/. (2017).

Jinjin Tang, Ying Song, and Xuesong Miller, Harveyand Zhou. 2016. 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* 66 (2016), 176–194.

Gwenola Thomas and Stéphane Donikian. 2000. Modelling virtual cities dedicated to behavioural animation. *Computer Graphics Forum* 19, 3 (2000), 71–80.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B* 67, 1 (2005), 91–108.

Uber. 2017. Uber Movement. movement.uber.com/. (2017).

Jur van den Berg, Jason Sewall, Ming Lin, and Dinesh Manocha. 2009. Virtualized traffic: Reconstructing traffic flows from discrete spatio-temporal data. In *Virtual Reality Conference, IEEE*. 183–190.

Hongling Wang, Joseph K Kearney, James Cremer, and Peter Willemsen. 2005. Steering behaviors for autonomous vehicles in virtual environments. In *Virtual Reality Conference, IEEE*. 155–162.

Yuqi Wang, Jiannong Cao, Wengen Li, and Tao Gu. 2016. Mining Traffic Congestion Correlation between Road Segments on GPS Trajectories. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*. 1–8.

Zuchao Wang, Tangzhi Ye, Min Lu, Xiaoru Yuan, Huamin Qu, Jacky Yuan, and Qianliang Wu. 2014. Visual exploration of sparse traffic trajectory data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1813–1822.

John Wardrop. 1952. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers* 1, 3 (1952), 325–362.

David Wilkie, Cenk Baykal, and Ming C Lin. 2014. Participatory route planning. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 213–222.

David Wilkie, Jason Sewall, Weizi Li, and Ming C. Lin. 2015. Virtualized Traffic at Metropolitan Scales. *Frontiers in Robotics and AI* 2 (2015), 11.

David Wilkie, Jason Sewall, and Ming C Lin. 2012. Transforming GIS data into functional road models for large-scale traffic simulation. *IEEE transactions on visualization and computer graphics* 18, 6 (2012), 890–901.

David Wilkie, Jason Sewall, and Ming C. Lin. 2013. Flow Reconstruction for Data-driven Traffic Animation. *ACM Trans. Graph.* 32, 4 (2013), 89:1–89:10.

David Wilkie, Jur P van den Berg, Ming C Lin, and Dinesh Manocha. 2011. Self-Aware Traffic Route Planning.. In *AAAI*, Vol. 11. 1521–1527.

P. Willemsen, J. K. Kearney, and H. Wang. 2006. Ribbon networks for modeling navigable paths of autonomous agents in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 331–342.

Daniel Work, Sébastien Blandin, Olli-Pekka Tossavainen, Benedetto Piccoli, and Alexandre Bayen. 2010. A traffic model for velocity data assimilation. *Applied Mathematics Research Express* 2010, 1 (2010), 1–35.

Cathy Wu, Jérôme Thai, Steve Yadlowsky, Alexei Pozdnoukhov, and Alexandre Bayen. 2015. Cellpath: Fusion of Cellular and Traffic Sensor Data for Route Flow Estimation via Convex Optimization. *Transportation Research Procedia* 7 (2015), 212 – 232.

Hai Yang, Tsuna Sasaki, Yasunori Iida, and Yasuo Asakura. 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B: Methodological* 26, 6 (1992), 417–434.

Xianfeng Yang, Yang Lu, and Wei Hao. 2017. Origin-Destination Estimation Using Probe Vehicle Trajectory and Link Counts. *Journal of Advanced Transportation* (2017).

Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. 2010. An interactive-voting based map matching algorithm. In *Proceedings of the 11th International Conference on Mobile Data Management*. 43–52.

Jia-Dong Zhang, Jin Xu, and Stephen Shaoyi Liao. 2013. Aggregating and sampling methods for processing GPS data streams for traffic state estimation. *IEEE Transactions on Intelligent Transportation Systems* 14, 4 (2013), 1629–1641.

Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. 2011. Urban computing with taxicabs. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 89–98.