2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Lagrangian Sensing: Traffic Estimation with Mobile Devices

Daniel B. Work, Olli-Pekka Tossavainen, Quinn Jacobson, and Alexandre M. Bayen

***Abstract*— An inverse modeling algorithm is developed to reconstruct the state of traffic (velocity field) on highways from GPS measurements gathered from mobile phones traveling on-board vehicles. The algorithm is based on *ensemble Kalman filtering* (EnKF), to overcome the nonlinearity and non-differentiability of a distributed highway traffic model for velocity. The algorithm is implemented in an architecture which includes GPS enabled phones and a privacy aware data collection infrastructure based on the novel concept of *Virtual Trip Lines* (a technology developed by Nokia). The data collection infrastructure is connected to a traffic estimation server running the EnKF algorithm online, and the estimation results are broadcast in real time back to mobile phones and to the internet. Results from the algorithm are presented using data collected during the February 8, 2008 *Mobile Century* experiment, in which a shock wave from a five-car accident is captured. A prototype estimation algorithm and system were run during the experiment, and highlight that measurements from as few as 2% to 5% of the commuting public are sufficient to accurately reconstruct the highway traffic state.**

I. INTRODUCTION

A. Smartphones as Lagrangian Sensors

With the standardization of GPS in mobile devices such as cell phones, and their increasing presence in vehicles in traffic, we are entering a new era of transportation system monitoring capabilities. The increased accuracy of GPS provides an appealing alternative to traditional approaches heavily relying on cell tower information, in particular triangulation and trilateration [1]. While tower monitoring approaches have proved to be useful to assess travel time for large spatial scales [2], [3] their performance for local traffic flow estimation in complex road network scales have been disappointing because of the lack of precision in the position and speed measurements. In contrast, as demonstrated with ongoing experiments [4], [5], GPS has the potential of making significant breakthroughs in highway traffic monitoring.

A fundamental challenge of using smartphone data (geo-referenced velocity) for highway traffic estimation is the development of a model for the evolution of traffic velocity. While GPS provides accurate speed measurements, accessing densities (on which most traffic flow models rely) from smartphones is currently not possible, because of the difficulty of creating empirical models capable of extrapolating the penetration rate of GPS equipped smartphones traveling in cars to vehicle density. This challenge is addressed by using a new model for the evolution of velocity, called CTM-v, in the form of a discrete time nonlinear dynamical system.

The second fundamental challenge in using smartphone data is the incorporation of *Lagrangian* measurements into a flow model. The term Lagrangian specifically refers to the fact that measurements are gathered from sensors which move along a trajectory in the field which is being sensed (the velocity field in the present case), rather than sampling at a fixed location. This is in contrast to *Eulerian* sensing, which refers to fixed sensors which collect measurements at predefined points. Classical traffic monitoring infrastructure relies on Eulerian sensors, for example loop detectors [1], [6], RFID transponders, radars and cameras [7].

The present article proposes a method capable of incorporating any Lagrangian velocity measurement in a velocity flow model. As will be explained later in the article, the current architecture on which the method is demonstrated produces measurements according to a specific privacy aware sampling procedure designed by Nokia. However, the proposed method works with any arbitrary sampling procedure (for example random sampling as in [8] or full trajectory sampling [9]).

B. Lagrangian Data Assimilation for Distributed Velocity Fields

The velocity field $v(x,t)$ on a highway segment $x \in [0,L]$ is a distributed parameter system in space. Vehicles labeled by $i \in \mathbb{N}$ travel along the highway with trajectories $x_i(t)$, and measure the velocity $v(x_i(t),t)$ along their trajectories (Lagrangian measurements). These measurements (discrete in time and space) are used to reconstruct or estimate the function $v(x,t)$, in a process referred to as *data assimilation* or *inverse modeling* [10]. Fig. 1 illustrates the process: the evolution of the velocity field $v(x,t)$ is a surface, which is to be reconstructed. A subset of the vehicles is sampled along their trajectories. For illustration purposes in the figure, four vehicles are sampled at time $t = t_m$, which produces four points on the $v(x,t)$ surface which can be used by the algorithm to reconstruct the surface.

Data from mobile devices can be obtained through a variety of sampling strategies, including a new paradigm developed by Nokia, called *Virtual Trip Lines* (VTLs), which act as virtual triggers for sensors on mobile platforms. The technique used to perform data assimilation with this sampling is described in Section II, which uses an algorithm

D. Work is a Ph.D. student in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720-1710 (e-mail: dbwork@berkeley.edu). Corresponding author.

O.-P. Tossavainen is a Post Doctoral researcher in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA, and with the Department of Physics, University of Kuopio, Kuopio, Finland.

Q. Jacobson is the Research Leader of the Mobile Internet Services Systems team, Nokia Research Center, Palo Alto, CA.

A. Bayen is an Assistant Professor in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

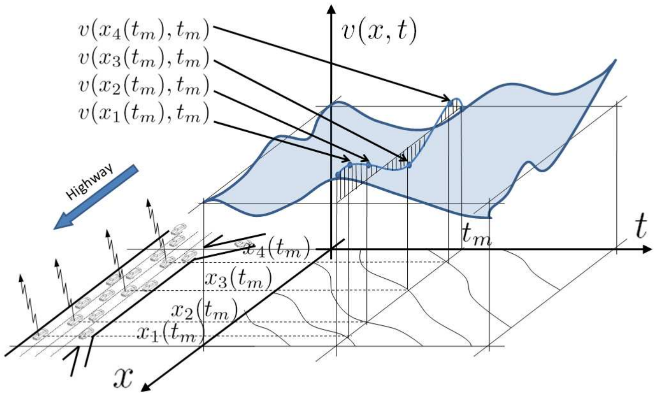

Fig. 1. Illustration of the distributed velocity field $v(x,t)$ to be reconstructed from Lagrangian samples. Four samples $v_i(x_i(t),t)$ are shown at $t = t_m$, from vehicles i transmitting their data (indicated by up-arrows above the vehicles).

based on *ensemble Kalman filtering* (EnKF). The section addresses the specific problem of data assimilation for the *Cell Transmission Model for velocity* (CTM-v), which is a velocity evolution model used in this article. Section III presents the system implementation used for gathering smartphone GPS data in a privacy aware environment, and addresses the data sampling mechanism (Section III-A) and the system architecture (Section III-B). Finally, Section IV presents the results from an unprecedented experiment realized on February 8, 2008, nicknamed *Mobile Century* for its 100 vehicles driving 10 mile loops for 8 hours, realizing a 2% to 5% penetration rate of equipped vehicles on the highway.

II. TRAFFIC ESTIMATION

A. Problem Statement

The goal of this article is to build an estimator to reconstruct the evolution of the velocity field on the highway. The highway transportation network is modeled as a directed graph consisting of vertices $\nu \in \mathcal{V}$ and edges $e \in \mathcal{E}$. Let L_e be the length of edge e. The spatial and temporal variables are $x \in [0, L_e]$, and $t \in [0, +\infty)$ respectively. In order to model traffic flow across the network, we define a junction $j \in \mathcal{J}$ as a tuple $\mathcal{J}_j := (\nu_j,\ I_j, \mathcal{O}_j) \subseteq \mathcal{V} \times \mathcal{E} \times \mathcal{E}$, consisting of a single vertex $\nu_j \in \mathcal{V}$, a set of incoming edges indexed by $e_{\text{in}} \in \mathcal{I}_j$, and a set of outgoing edges indexed by $e_{\text{out}} \in \mathcal{O}_j$. The objective is to estimate the velocity field at discrete points $i = 0$ to $i = i_{\max}$ in space at each time step n denoted by: $v^n := \left[v^n_{0,e_0}, \cdots, v^n_{i_{\max},e_0}, \cdots, v^n_{0,|\mathcal{E}|}, \cdots, v^n_{i_{\max},|\mathcal{E}|}\right]$ for all edges $e \in \mathcal{E}$ in the network, using velocity data obtained from the mobile devices.

B. Related Work

Kalman filtering (KF) has been widely used for traffic state estimation in earlier studies in its various forms. In [11], *mixture Kalman filtering* (MKF) was applied to the *Cell Transmission Model* (CTM) [12] to estimate traffic densities for ramp metering. The nonlinear CTM was transformed into a switching state space model, which enabled the use of a set of linear equations to describe the state evolution for the distinct flow regimes on the highway (e.g. highway is in free-flow or congestion). In [7], a Kalman filter was used to incorporate Lagrangian velocity trajectories into a density based CTM for highway traffic. A real–time algorithm for traffic estimation based on the *extended Kalman filter* (EKF) using a second order flow model was used in [13]. A key ingredient of this work is the differentiability of the numerical scheme employed for the second order model of traffic used by the authors, a feature which our model does not possess. Other treatments of traffic estimation include adjoint based control and data assimilation in [14], [15], *unscented Kalman filtering* (UKF) in [16] and *particle filtering* (PF) in [16], [17], [18].

A common feature for CTM based methods [7] described above is that the evolution of traffic state (typically density, not velocity) relies on a set of linearized equations which are needed in order to use the KF or EKF techniques. On the other hand, the PF technique is a nonlinear scheme for solving the Bayesian update problem, but has a higher computational cost. The approach proposed in the present work employs *ensemble Kalman filtering* (EnKF) [19], which enables the use of fully nonlinear evolution equations such as the discretization of the new flow model implemented in this article, while exploiting its linear observation equation. Unlike UKF, which uses a deterministic sampling technique, EnKF uses Monte Carlo integrations to maintain the nonlinear features of error statistics. Furthermore, by employing a fully nonlinear velocity evolution model, no highway mode selection algorithms or simplifications to the equations are needed in this work.

Earlier studies have specifically approached the highway traffic estimation problem using cell phone network data. In [20], an EKF was applied to a second order model of vehicle density and velocity, and validated in simulation. In practice, the modeling assumption that network providers can accurately provide both density and flow of the cellular phones currently on the highway of interest is limiting, especially in dense roadway networks. The work [21] uses a fully nonlinear particle filter to assimilate the mean velocity of a vehicle traveling between cell tower hand-off points, but also suffers from the same practical limitations in dense road networks.

C. Kalman Filtering

1) State–Space Model: Given the velocity field at all points on the network at time $n\Delta T$, the velocity at time $(n+1)\Delta T$ is constructed using the CTM-v algorithm denoted by: $v^{n+1} = \mathcal{M}[v^n]$. This algorithm consists of the following steps. For each vertex in the network, a linear program is solved such that strong boundary conditions are imposed on the incoming and outgoing edges of the junction [22]. Next, the velocity field at the i^{th} discrete point on each edge is updated according to a nonlinear non-differentiable numerical scheme resulting from a first order scalar hyperbolic partial differential equation, which has been transformed from vehicle density to velocity:

$$v_i^{n+1} = V\left(V^{-1}\left(v_i^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_i^n, v_{i+1}^n\right) - \tilde{G}\left(v_{i-1}^n, v_i^n\right)\right)\right) \tag{1}$$

where the velocity flux $\tilde{G}(v_1, v_2)$ is given by:

$$\tilde{G}(v_1, v_2) = \begin{cases} V^{-1}(v_2)\, v_2 & \text{if } v_c \geq v_2 \geq v_1 \\ V^{-1}(v_c)\, v_c & \text{if } v_2 \geq v_c \geq v_1 \\ V^{-1}(v_1)\, v_1 & \text{if } v_2 \geq v_1 \geq v_c \\ \min\big(V^{-1}(v_1)\, v_1, & \\ \qquad V^{-1}(v_2)\, v_2\big) & \text{if } v_1 \geq v_2 \end{cases} \tag{2}$$

with an invertible velocity function:

$$v = V(\rho) = \begin{cases} v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right) & \text{if } \rho \leq \rho_c \\ -w_f\left(1 - \frac{\rho_{\max}}{\rho}\right) & \text{otherwise} \end{cases} \tag{3}$$

In the equations above, $v_{\max}$ (respectively $\rho_{\max}$) is the maximum velocity (density) on the roadway, v_c (ρ_c) is the critical velocity (density) where the highway transitions from freeflow to congestion, and w_f is the maximum congestion propagation speed.

For estimation purposes, we extend the model to

$$v^n = \mathcal{M}[v^{n-1}] + \eta^n \tag{4}$$

where $\eta^n \sim (0, \mathbf{Q}^n)$ is the Gaussian zero-mean, white state noise with covariance $\mathbf{Q}^n$.

A network observation model is given by:

$$y^n = \mathbf{H}^n v^n + \chi^n \tag{5}$$

The linear observation matrix $\mathbf{H}^n \in \{0,1\}^{p^n \times \kappa}$ encodes the p^n discrete cells on the highway for which the velocity is observed during discrete time step n and $\kappa = \sum_{e \in \mathcal{E}} (i_{\max,e} + 1)$ is the corresponding (total) number of cells in the network. The last term in expression (5) is the white, zero mean observation noise $\chi^n \sim (0, \mathbf{R}^n)$ with covariance matrix $\mathbf{R}^n$.

2) Extended Kalman Filtering For Nonlinear Systems: Two fundamental challenges arise when sensing traffic conditions using mobile phones, which appear in the observation model. The first challenge (which motivates the early parts of this article) is that for density–based models, the state is not observed directly. In particular, when applying the Daganzo–Newell (triangular) velocity function (common in the transportation literature), both the dynamical system which describes the density evolution and the observation model which characterizes the observations are nonlinear. The nonlinear observation operator must be linearized at the cost of accuracy or overcome with computationally intensive particle filtering techniques. Instead, by developing equivalent models in which the velocity is stored as the state, the observation operator becomes linear. The second challenge is a consequence of the motion of the sensor. The observation model must capture the Lagrangian nature of the sensors, whose motion is coupled with the model itself by integration of the velocity field. Because of the high accuracy of the GPS position measurements, the location of the observation can be used to construct the observation operator a posteriori (*i.e.* reduced to a time-varying observation matrix).

If equation (1) was differentiable in v^n, so would be the operator $\mathcal{M}[\cdot]$ in (4), in which case the optimal estimate for the state v^n could be obtained using the following traditional equations known as the extended Kalman filter:

- Forecast step (Time-update):

$$v_f^n = \mathcal{M}[{v_a}^{n-1}]$$
$$\mathbf{P}_f^n = \mathcal{M}_L^{n-1}\mathbf{P}_a^{n-1}\left(\mathcal{M}_L^{n-1}\right)^T + \mathbf{Q}^{n-1} \tag{6}$$

  where $\mathcal{M}_L$ is the Jacobian matrix of mapping $\mathcal{M}$ (also known as the *tangent linear model*) defined as

$$\mathcal{M}_L^{n-1}(i,j) = \frac{\partial \mathcal{M}_i[v_a^{n-1}]}{\partial v_j^n} \tag{7}$$

- Analysis step (Measurement-update):

$$v_a^n = v_f^n + \mathbf{G}^n\left(y^n - \mathbf{H}^n v_f^n\right) \tag{8}$$
$$\mathbf{P}_a^n = \mathbf{P}_f^n - \mathbf{G}^n\mathbf{H}^n\mathbf{P}_f^n \tag{9}$$
$$\mathbf{G}^n = \mathbf{P}_f^n\left(\mathbf{H}^n\right)^T\left(\mathbf{H}^n\mathbf{P}_f^n\left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{10}$$

  where $\mathbf{P}_f^n$ (resp. $\mathbf{P}_a^n$) is the error covariance of the forecast (analyzed) state at time n.

The initial conditions for the recursion are given by $v_a^0 = v^0$ and $\mathbf{P}_a^0 = \mathbf{P}^0$.

3) Ensemble Kalman Filter: The ensemble Kalman filter was introduced by Evensen in [19] as an alternative to EKF to overcome specific difficulties with nonlinear state evolution models, including non-differentiability of the model and closure problems. Closure problems refer to the fact that in EKF, it is assumed that discarding the higher order moments from the evolution of the error covariance in (6) yields a good approximation. However, in cases in which this linearization approximation is invalid, it can cause an unbounded error variance growth [19]. To tackle this issue EnKF uses Monte Carlo (or ensemble integrations). By propagating the ensemble of model states forward in time, it is possible to calculate the mean and the covariances of the error needed at analysis (measurement-update) steps [23] and avoid the closure problem. Furthermore, a strength of EnKF is that it uses the standard update equations of EKF, except that the gain is computed from the error covariances provided by the ensemble of model states.

EnKF also comes with a relatively low numerical cost. Namely, usually a rather limited number of ensemble members is needed to achieve a reasonable statistical convergence [23].

In traditional Kalman filtering, the error covariance matrices are defined in terms of the true state as $\mathbf{P}_f = E[(v_f - v_t)(v_f - v_t)^T]$ and $\mathbf{P}_a = E[(v_a - v_t)(v_a - v_t)^T]$ where $E[\cdot]$ denotes the average over the ensemble, v is the model state vector at particular time, and the subscripts f, a, and t represent the forecast, analyzed, and true state, respectively. However, since the true state is not known, ensemble covariances for EnKF have to be considered. These covariance matrices are evaluated around the ensemble mean

$\bar{v}$, yielding $\mathbf{P}_f \approx \mathbf{P}_{\text{ens},f} = E[(v_f - \bar{v}_f)(v_f - \bar{v}_f))^T]$ and $\mathbf{P}_a \approx \mathbf{P}_{\text{ens},a} = E[(v_a - \bar{v}_a)(v_a - \bar{v}_a)^T]$ where the subscript ens refers to the ensemble approximation. In [23], it is shown that if the ensemble mean is used as the best estimate, the ensemble covariance can consistently be interpreted as the error covariance of the best estimate. For complete details of derivation of the EnKF algorithm, the reader is referred to [19].

The ensemble Kalman filter algorithm can be summarized as follows [19], [23]:

1) *Initialization*: Draw K ensemble realizations $v_a^0(k)$ (with $k \in \{1, \cdots, K\}$) from a process with a mean speed $\bar{v}_a^0$ and covariance $\mathbf{P}_a^0$.
2) *Forecast*: Update each of the K ensemble members according to the CTM-v (4) forward simulation algorithm. Then update the ensemble mean and covariance according to:

$$v_f^n(k) = \mathcal{M}[v_a^{n-1}(k)] + \eta^n(k). \tag{11}$$

$$\bar{v}_f^n = \frac{1}{K}\sum_{k=1}^{K} v_f^n(k). \tag{12}$$

$$\mathbf{P}_{\text{ens},f}^n = \frac{1}{K-1}\sum_{k=1}^{K}\left(v_f^n(k) - \bar{v}_f^n\right)\left(v_f^n(k) - \bar{v}_f^n\right)^T \tag{13}$$

3) *Analysis*: Obtain measurements, compute the Kalman gain, and update the network forecast:

$$\mathbf{G}_{\text{ens}}^n = \mathbf{P}_{\text{ens},f}^n\left(\mathbf{H}^n\right)^T\left(\mathbf{H}^n\mathbf{P}_{\text{ens},f}^n\left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{14}$$

$$v_a^n(k) = v_f^n(k) + \mathbf{G}_{\text{ens}}^n\left(y_{\text{meas}}^n - \mathbf{H}^n v_f^n(k) + \chi^n(k)\right) \tag{15}$$

4) Return to 2.

In (15), an important step is that at measurement times, each measurement is represented by an ensemble. This ensemble has the actual measurement as the mean and the variance of the ensemble is used to represent the measurement errors. This is done by adding perturbations $\chi^n(k)$ to the measurements drawn from a distribution with zero mean and covariance equal to the measurement error covariance matrix $\mathbf{R}^n$. This ensures that the updated ensemble has a variance that is not too low [23].

4) Large Scale Real–Time Implementation: The ensemble Kalman filter algorithm presented in the previous section is in a framework in which all of the unknown state variables on each edge in the network are updated simultaneously. This introduces the following problems. First, because the state covariance is represented through a limited number of ensemble members, non-physical correlations may arise. This means that the correlation matrix may incorrectly show correlation between distant parts of the highway network which do not correlate in practice. Secondly, the framework described previously requires the forecast error covariance in (13) to be computed for the entire highway network, then used for computing the Kalman gain in (14). When operating on large scale networks such as the San Francisco Bay Area, CA, the covariance matrix can easily require more than 2 GB of memory to load, creating computational limitations for implementation.

To circumvent the above mentioned problems for practical implementation, we employ a *covariance localization method*. This approach limits the correlation between the velocity states on all edges in the network. For a given edge e, only nearby links (upstream and downstream in the network) can exhibit correlation, thereby removing correlation across distant parts of the network. These techniques have also been implemented for oceanography data assimilation problems (see e.g. [24]).

For this large scale traffic network estimation problem, localization also provides a computationally efficient way to update the state variables at the measurement update time in (14)–(15). Namely, due to the localization, the computation of the covariance matrix in (13) is transformed into a computation of numerous small localized covariance matrices for each edge in the network. These small scale covariance matrices are computed for each edge given its neighboring edges on which the correlation is assumed to be physically meaningful. Finally, this allows the distributed solving of the update equations.

For the localization, we introduce a localization operator $\mathcal{L}_e$ for each edge e, which is constructed at the initialization stage. This operator indicates which velocity states on the other edges of the network are allowed to have correlation with the velocity state on the eth edge. The implementation of the EnKF algorithm described previously can be modified for localization by replacing the measurement update equations (13)-(15) with the following sub-algorithm:

For each edge $e \in \mathcal{E}$:

1) Using the localization operator $\mathcal{L}_e$, compute the localized forecast error covariance:

$$\mathbf{P}_{\text{ens},f,e}^n = \frac{1}{K-1}\sum_{k=1}^{K}\mathcal{L}_e\left(v_f^n(k) - \bar{v}_f^n\right) \times \left(\mathcal{L}_e(v_f^n(k) - \bar{v}_f^n)\right)^T \tag{16}$$

2) *Analysis*: Obtain measurements $y_{\text{meas},e}^n$ from edges that are indicated in $\mathcal{L}_e$, compute the Kalman gain, and update the the local forecast:

$$\mathbf{G}_{\text{ens},e}^n = \mathbf{P}_{\text{ens},f,e}^n\left(\mathbf{H}_e^n\right)^T \times \left(\mathbf{H}_e^n\mathbf{P}_{\text{ens},f,e}^n\left(\mathbf{H}_e^n\right)^T + \mathbf{R}_e^n\right)^{-1} \tag{17}$$

$$v_{a,e}^n(k) = \mathcal{L}_e\left(v_f^n(k)\right) + \mathbf{G}_{\text{ens},e}^n\left(y_{\text{meas},e}^n - \mathbf{H}_e^n v_f^n(k) + \chi_e^n(k)\right) \tag{18}$$

3) Return to 1.

It is worth noting that in practice, the operator $\mathcal{L}_e$ does not need to be constructed as a matrix in the computer memory and subsequently be used to do the relatively demanding matrix multiplications. In other words, the eth edge has references to the forecasts and measurements of its neighboring edges needed to construct the localized forecast error covariance matrix.

III. SYSTEM IMPLEMENTATION

This section presents the system jointly developed by Nokia and UC Berkeley to estimate traffic using GPS-equipped cell phones. From a system design perspective, the main challenges with data collection using GPS-equipped smartphones arise from the fact that unlike conventional traffic monitoring infrastructure, the phone is not a dedicated sensor. In order to maintain a functioning system, with humans carrying these devices, resource consumption such as battery and bandwidth must be sufficiently low, and some degree of privacy and anonymity should be maintained, for the driving public to opt in. These issues have a major impact on the sampling techniques used for collecting measurements.

A. Sampling and Data Collection

A variety of sampling techniques can be used to collect data from GPS enabled Lagrangian sensors. In the case of the Nokia N95, the embedded GPS chipset is capable of producing a geo-position (latitude, longitude, altitude) every three seconds. A simple filter is implemented to produce an estimate of the velocity at the same frequency. Over time, this vehicle trajectory and velocity information produces a rich history of the dynamics of the vehicle and the velocity field through which it evolves.

While this level of detail is particularly useful for traffic estimation, it can be extremely privacy invasive, since the device is ultimately identifiable with a single user. Even if personally identifiable information is replaced with a randomly chosen identification, it is still possible to re-identify individuals. For example, pseudoanonymous trajectories have been combined with free, publicly available data sets to determine the locations of participants' homes [25].

The transmission of high frequency data without regard to location wastes resources throughout the system. In addition to disclosing sensitive information, the trajectory information on small roadways near users homes are of lower value to the general commuting public than major thoroughfares such as interstates. Thus, collection of low utility and highly sensitive data should be avoided.

A variety of methods can be used to address these problems. To manage privacy concerns, in addition to pseudoanonymization of the trajectory data, the data can be degraded until a sufficient level of privacy is attained. Common degradation approaches include (i) spatial obfuscation (*i.e.* blocking data collection from particular regions, such as home), (ii) increasing uncertainty through noise addition, and (iii) location discretization approaches which round the measurement to the nearest discrete grid point. The trade-offs between the measurement utility and privacy under these degradation approaches have been analyzed with experimental data [26] and can be cast as a sampling strategy optimization problem [8].

An alternative sampling strategy which is implemented in this work is based on *Virtual Trip Lines* (VTLs) [27], which act as spatial triggers for phones to send updates. Each VTL is composed of two GPS coordinates which make a virtual line drawn on a roadway of interest. Instead of periodic sampling in time, VTLs control disclosure of speed and location updates by sampling in space, creating updates at predefined geographic locations on roadways of interest.

Mobile devices monitor their speed and location using GPS and use the locally stored VTLs to determine when a VTL crossing occurs. When the phone intersects a VTL, the device can probabilistically send an update to a back end server with anonymized position, speed and direction information. The device may also probabilistically send the travel time observed between two consecutive trip lines.

A unique feature of this sampling strategy is that data points are only identified through the ID of the VTL, and not that of the mobile device which generated the update, so no extended trajectories are collected. Through careful placement of trip lines, the system is better suited to manage data quality and privacy than through a uniform temporal sampling interval.

B. System Architecture

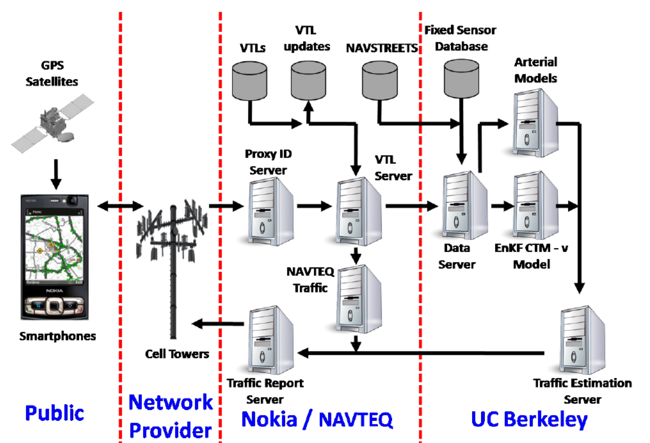

Fig. 2. System architecture overview. The system consists of vehicles equipped with GPS-enabled smartphones, a cellular network provider, an ID data collection infrastructure and traffic information provision, and traffic estimation algorithm running on the model server.

The resulting system architecture which supports this research (shown in Fig. 2) consists of four layers: GPS-enabled smartphones in vehicles (driving public), a cellular network operator (network provider), cellular phone data aggregation and traffic service provision (Nokia/Navteq), and traffic estimation (Berkeley). On each participating mobile device (or client), an application is executed (see Fig. 3) which is responsible for the following functions: downloading and caching trip lines from the VTL server, detecting trip line traversal, and filtering measurements before transmissions to the service provider. To determine trip line traversals, probe vehicles check if the line between the current GPS position and the previous GPS position intersects with any of the trip lines in its cache. Upon traversal, the mobile device creates an encrypted VTL update. The update includes speed readings, timestamps, the tripline ID, and direction trip line crossing. These VTL updates are transmitted to the ID proxy server over a secure channel.

Fig. 3. Nokia N95 client emulator used for development of our system. The client serves both to collect data from individual devices, and to display current traffic conditions. The colors in the circles represent the speed of traffic (green corresponds to free flow; red is congested). (Left) Free flowing traffic in Santa Clara, CA. (Middle) Congestion occurs in San Francisco, CA. (Right) Bay Area regional traffic.

Note that all data packets transmitted from the mobile device must contain the mobile device identification information which is provided to the communication provider for billing purposes. Thus, an ID proxy server is used to first authenticate each client to prevent unauthorized updates, then remove the mobile device identification information from the data packets. It then forwards anonymized updates to the VTL server. Since the VTL update is encrypted with the VTL server's public key (RSA encryption), the ID proxy server cannot access the VTL update content. It only has knowledge of which phone transmitted a VTL update, but no knowledge of the phone's position or speed information. Thus we prevent any single entity from observing both the identification data required by the network operator, and the sensing data. See [27] for a more detailed description of privacy protection in VTL based traffic monitoring.

The VTL server stores all trip lines in a VTL database and distributes trip lines within a given region to a mobile device upon receiving a VTL download request for that region. The VTL server also aggregates updates from a large number of probe vehicles in the VTL update database and pushes the data to the UC Berkeley data server for data assimilation, which combines the cell phone data with other data such as loop detectors. The data server also provides the Navteq (now owned by Nokia) NAVSTREETS digital map data to the EnKF model. This data is essential since it contains the roadway geometries upon which the base network for the EnKF algorithm is constructed.

In addition to the EnKF algorithm described in this work, numerous other estimation algorithms are run in parallel as part of ongoing research, including arterial traffic models. An estimate manager in the traffic estimation server monitors the performance of the various algorithms and transmits the results to the traffic report server. The estimates are integrated with estimates from traffic models provided by Navteq before being transmitted back to the mobile device.

The current VTL implementation generates approximately 1KB of update data for every two minutes per client while driving on a major road. Assuming an average two hours of driving per day on a major road, we expect the total data transfer is 60KB per day. The database servers can easily scale to large number of client updates since the bandwidth and the total data storage demands are rather small by current information industry standards.

IV. EXPERIMENTAL RESULTS

A. Mobile Century Case Study (February 8, 2008)

Nicknamed the *Mobile Century* experiment, a prototype privacy-aware data collection system was launched on February 8, 2008 and used to estimate traffic conditions for a day on I-880 near San Francisco, CA. With the help of 165 UC Berkeley students, 100 vehicles carrying Nokia N95 phones drove repeated loops of six to ten miles in length continuously for eight hours. These vehicles represented approximately 2% to 5% of the total volume of traffic on the main line of the highway during the experiment.

This section of highway was selected specifically for its complex traffic properties, which include alternating periods of free-flowing, uncongested traffic, and slower moving traffic during periods of heavy congestion. The section is also covered with existing loop detectors feeding into the PeMS system [28], which are used to assess the quality of the EnKF estimates.

B. Implementation and Results

The network implemented for the results presented in this article is a 7 mile stretch of I880 northbound from the Decoto Rd. entrance ramp (south end), to the Winton Ave. exit ramp (north end). The network model consists of 13 edges and 14 junctions (6 exit ramps, 7 entrance ramps, and one lane drop). A total of 40 VTLs were placed on this highway segment with an average spacing of 0.17 miles.

At approximately 10:30 am, a multiple car accident created significant unanticipated congestion for northbound traffic south of CA 92 (see Fig. 4). An earlier version of the EnKF algorithm, runnionng in real-time during the experiment, detected the accident's resulting bottleneck and corresponding shock wave [4]. It broadcast the speed contour of the highway and the resulting congestion in real time [29]. In Figs. 5-8, we present a comparison of the velocity estimate from the EnKF CTM-v algorithm using VTL data only with the velocity estimate obtained from the PeMS system [30], which provides loop detector data for the deployment area and serves as benchmark for this method.

In general, the results of the EnKF estimation show good agreement with the PeMS velocity estimate. In particular, the VTL-based sensing coupled with the EnKF algorithm captures the main features of the congestion pattern, including the length of the resulting queue, which extends just over two miles at 10:52 am (see Figs. 5 and 6). This proof of concept is an important step forward in mobile device-based sensing

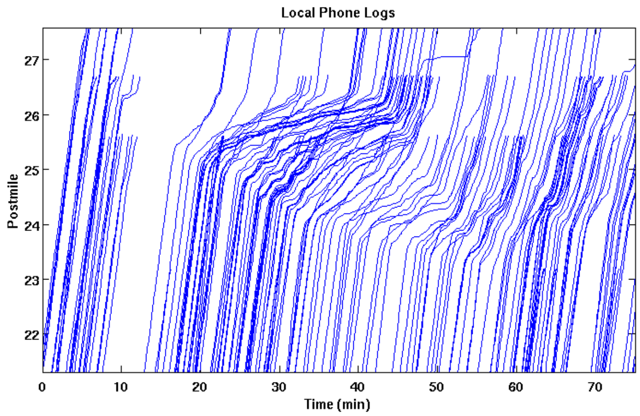

Fig. 4. Local phone logs, I880N, Feb. 8, 2008. In addition to the VTL updates, the raw trajectory of each device was recorded locally to the device as a backup for the data collection infrastructure for the purposes of this experiment only. The sharp decrease in the slope (velocity) of the trajectory corresponds to the vehicle encountering the shockwave and entering congestion. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top). Trajectories are in the direction of increasing y.

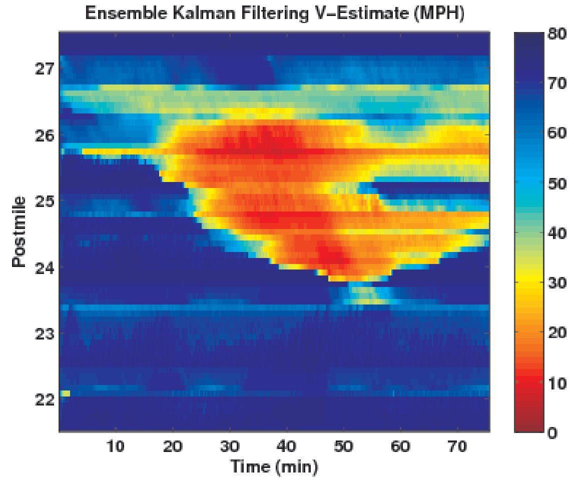

Fig. 5. EnKF velocity contour plot, I880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

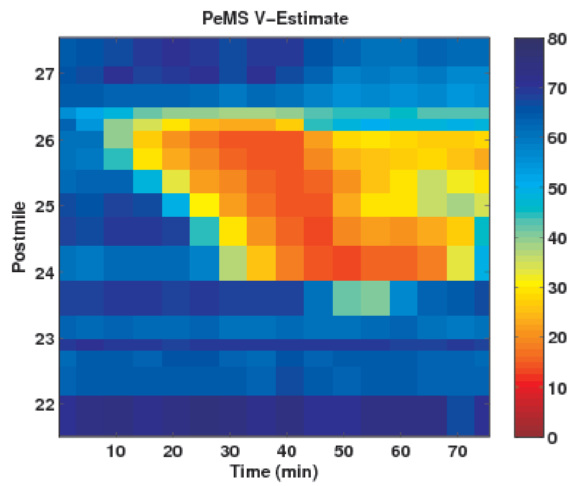

Fig. 6. PeMS velocity contour plot, I880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

because of the sparsity of data used for the EnKF estimate. Unlike the loop detectors which sense every vehicle in each lane on the highway, but at fixed points in space, the mobile device-based sensing collects data from a very small fraction of vehicles. Furthermore, because of privacy considerations, the vehicles are not tracked in space; only a subset of the data logged by each device is used for estimation, sampling only anonymous location and speed updates triggered by VTLs. No extended vehicle-trajectory travel times are collected or used for estimation.

Note that there are some differences in the speed estimation shown in Figs. 5 and 6. In Fig. 7, the relative difference between the EnKF and PeMS contour is shown, with EnKF as the reference. In the free flowing regions, the relative difference is quite small. The absolute speed difference in this regime is shown with a dashed green line in Fig. 8 for a sample postmile of 22.8. As expected, the spikes in high relative difference in Fig. 7 occur in the queue resulting from the accident. The postmile with the greatest magnitude relative difference (PM 24.6, with absolute speed difference plotted as a dash dot red line in Fig. 7) occurs because of two factors. First, the EnKF estimates the velocity contour at a temporal resolution on the order of seconds, while the PeMS estimate shown is aggregated over a five minute window. Second, because the absolute speed in the congested regime is small, any difference in speed is amplified. Ultimately, the difference between PeMS and the EnKF on average is less than 10% across the network, which highlights the potential utility of mobile devices as a source of traffic data in the future.

V. CONCLUSION AND FUTURE WORK

In this article, a new traffic data collection paradigm using GPS-equipped mobile devices was implemented using a privacy aware architecture. A nonlinear time invariant dynamical system forms the basis of the ensemble Kalman filtering algorithm, which is introduced because of the nonlinearity and non-differentiability of the model. The algorithm was validated using data obtained from the *Mobile Century* field experiment, and shows good agreement with PeMS loop detector data, even at penetration rates below five percent. This algorithm will be implemented next for a live system in which both fixed loop detector data and cell phone data is fused to produce traffic estimates in Northern California as part of a follow-up field operational test known as *Mobile Millennium* [29].

ACKNOWLEDGMENTS

The authors wish to thank Ken Tracton, Toch Iwuchukwu, Dave Sutter, and Murali Annavaram at Nokia Research Center Palo Alto, Baik Hoh and Marco Gruteser of Winlab at Rutgers University, and Jeff Ban, Ryan Herring, Juan Carlos Herrera, Christian Claudel, and Sebastien Blandin for their invaluable contributions to develop, build, and deploy the traffic monitoring system implemented as part of the Mobile Century experiment. We thank the staff of the California Center for Innovative Transportation

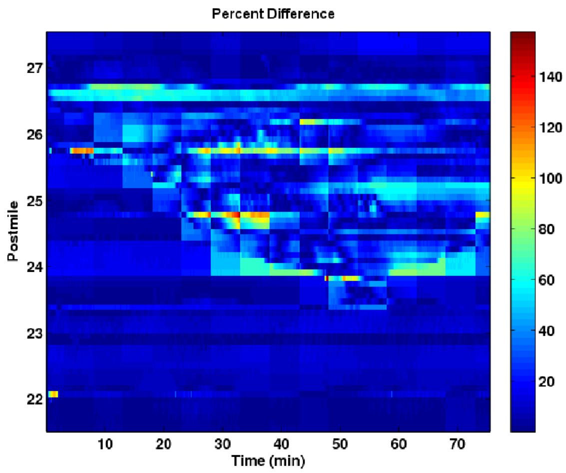

Fig. 7. PeMS-EnKF relative difference plot, I880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

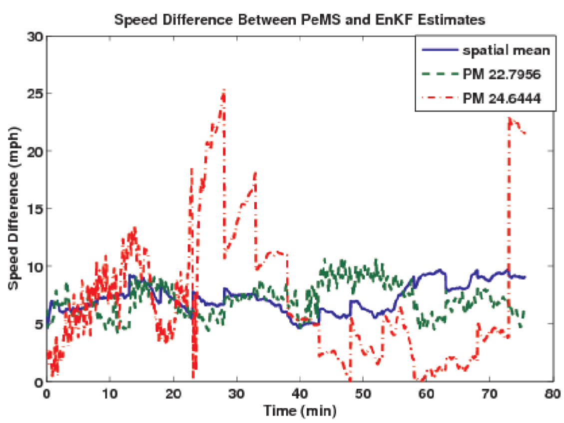

Fig. 8. PeMS-EnKF absolute speed difference, I880N, Feb. 8, 2008. Absolute difference in the velocity estimates between EnKF and PeMS spatially averaged across the network (solid blue), at postmile 22.8 with low relative error (dashed green), and postmile 24.6 with high relative error (dash dot red) as a function of time. x-axis: time in minutes past 10:12am. y-axis: absolute speed difference between EnKF and PeMS.

for the Mobile Century logistics planning and successful implementation. This research was supported by the Federal and California DOTs, Nokia, the Center for Information Technology Research in the Interest of Society, the Finnish Funding Agency for Technology and Innovation (Tekes), the National Science Foundation under contract CNS-0615299, and the US Department of Transportation through the Dwight David Eisenhower Transportation Fellowship Program.

References

[1] B. Smith and M. Fontaine, "Private-sector provision of congestion data." Washington, D.C.: Transportation Research Board, 2006, NCHRP Project 70-01.

[2] 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 Part C*, vol. 15, no. 6, pp. 380–391, 2007.

[3] M. González, C. Hidalgo, and A. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[4] D. Work, O.-P. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," in *Proc. of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008, pp. 2141–2147.

[5] S. Amin et al., "Mobile century-using GPS mobile phones as traffic sensors: a field experiment," in 15^{th} *World congress on ITS*. New York, N.Y.: Intelligent Transport Systems, November 16-20 2008.

[6] Z. Jia, C. Chen, B. Coifman, and P. Varaiya, "The PeMS algorithms for accurate, real time estimates of g-factors and speeds from single loop detectors," in *IEEE Intelligent Transportation Systems Conference Proceedings*, Oakland, CA, Aug. 2001, pp. 536–541.

[7] J.-C. Herrera and A. Bayen, "Traffic flow reconstruction using mobile sensors and loop detector data," in 87^{th} *TRB Annual Meeting*. Washington D.C.: Transportation Research Board, Jan. 12-17 2008.

[8] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *IPSN 2008, International Conference on Information Processing in Sensor Networks*, St. Louis, MI, Apr. 2008.

[9] A. Schwarzenegger, D. E. Bonner, W. Kempton, and R. Copp, "State highway congestion monitoring program (HICOMP), annual data compilation," Caltrans, Tech. Rep., June 2008.

[10] J. M. Lewis, S. Lakshmivarahan, and S. Dhall, *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge, UK: Cambridge University Press, 2006.

[11] X. Sun, L. Munoz, and R. Horowitz, "Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application," in *Proc. of the American Control Conference*, vol. 3, Boston, MA, 2004, pp. 2098 – 2103.

[12] C. F. Daganzo, "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B*, vol. 28, no. 4, pp. 269–287, 1994.

[13] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: a general approach," *Transportation Research Part B*, vol. 39, no. 2, pp. 141–167, 2005.

[14] D. Jacquet, C. Canudas de Wit, and D. Koenig, "Traffic control and monitoring with a macroscopic model in the presence of strong congestion waves," in *Proc. of the 44th IEEE Conference on Decision and Control, and European Control Conference*, Sevilla, Spain, 2005, pp. 2164–2169.

[15] D. Jacquet, M. Krstic, and C. Canudas de Wit, "Optimal control of scalar one-dimensional conservation laws," in *Proc. of the 25th American Control Conference*, Minneapolis, MN, 2006, pp. 5213–5218.

[16] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within recursive bayesian framework," *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.

[17] J. Sau, N. El Faouzi, A. Ben Assa, and O. De Mouzon, "Particle filter-based real-time estimation and prediction of traffic conditions," *Applied Stochastic Models and Data Analysis*, vol. 12, 2007.

[18] A. Hegyi, L. Mihaylova, R. Boel, and Z. Lendek, "Parallelized particle filtering for freeway traffic state tracking," in *Proc. of the European Control Conference*, Kos, Greece, July 2007, pp. 2442–2449.

[19] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*. Berlin Heidelberg: Springer-Verlag, 2007.

[20] A. Alessandri, R. Bolla, and M. Repetto, "Estimation of freeway traffic variables using information from mobile phones," in *Proc. American Control Conference the 2003*, vol. 5, Denver, CO, June 2003, pp. 4089–4094.

[21] P. Cheng, Z. Qiu, and B. Ran, "Particle filter based traffic state estimation using cell phone network data," in *Proc. IEEE Intelligent Transportation Systems Conference ITSC '06*, 2006, pp. 1047–1052.

[22] M. Garavello and B. Piccoli, *Traffic Flow on Networks*, ser. American Institute of Mathematical Sciences on Applied Math, A. Bressan, Ed., Springfield, MO, 2006.

[23] G. Burgers, P. Jan van Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly Weather Review*, vol. 126, no. 6, pp. 1719–1724, 1998.

[24] H. Mitchell, P. Houtekamer, and G. Pellerin, "Ensemble size, balance, and model–error representation in an ensemble Kalman filter," *Montly Weather Review*, vol. 130, pp. 2791–2808, 2002.

[25] 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, 2006.

[26] J. Krumm, "Inference attacks on location tracks," in *Fifth International Conference on Pervasive Computing (Pervasive 2007)*, Toronto, Ontario, Canada., May 2007.

[27] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *6th International Conference on Mobile Systems, Applications, and Services*, Breckenridge, CO, June 17-18 2008, pp. 15–28.

[28] C. Chen, P. Varaiya, and J. Kwon, "An empirical assessment of traffic operations," in *International Symposium on Transportation and Traffic Theory*, College Park, MD, July 2005, pp. 19–21.

[29] `http://traffic.berkeley.edu/`.

[30] `http://pems.eecs.berkeley.edu/`.