Sensor for Unexpected Roadway Events: Field Trials (SURE-FT) Final Report

METRANS Project DTRS98-G-0019

John Heidemann* (PI) , Genevieve Guiliano†(co-PI), Unkyu Park*, Fabio Silva*
Mengzhao Hu†, Xi Wang*, Nikhil Prashar†
USC/Information Sciences Institute
†USC/School of Policy, Planning, and Development

Final Report: January 8, 2008
(draft final report: January 19, 2007)

Disclaimer

Abstract

The purpose of this research is to demonstrate the feasibility of an automated system for medium-duration, deployable, in-the-field vehicle classification. Most temporary surveys of vehicle traffic today are done manually, typically with human observers recording traffic. Instead, we plan to employ a network of traffic sensors (NOTS), a number of small, low-cost computer nodes, each with a portable inductive loop sensor. This system can provide accurate measurements for longer duration and lower cost than is possible today. Our system will allow medium-term surveys, targeting 7-14 days. For temporary surveys like these it is essential that the system be easily deployable, not requiring investment in permanent, in-roadway sensors, but instead capable of being deployed by one or a few people in a few hours. We look at accurate vehicle classification and compare our results to ground truth taken from off-line analysis of videos, and to on-line human observations.

Contents

List of Figures

List of Tables

1 Introduction

This report summarizes the research done as part of the Sensor for Unexpected Roadway Events: Field Trials (SURE-FT) project, from January 2005 through August 2006.

SURE-FT is applying sensor networks to classify vehicle traffic. We are developing a network of traffic sensors (NOTS), which are small, low-cost computer nodes, each with a portable inductive loop sensor. A prior project, SURE-SE project [15], defined this problem, reviewed related work and the current state-of-the-art (summarized in Section 2), and conducted a preliminary experiment at USC to collect data (described in Section 5.1).

Our goal in SURE-FT was to develop algorithms to classify vehicles by type (using the FHWA classification system [5]), to evaluate the use of multiple sensors to reduce errors, and to integrate data collection, configuration, and processing software.

The remainder of this introduction reviews the need this research (Section 1.1), limitations of current approaches (Section 1.2). The remainder of the report reviews prior work (Section 2), describes our proposed system (Section 3), the algorithms we developed for single- and multi-node classification (Section 4), prior and new data collection (Section 5), and evaluation of these algorithms (Section 6).

1.1 Background and Justification of Research

There is an ongoing need for traffic data to validate and calibrate regional and local transportation models. Regional models are used to evaluate alternative short and long range transportation plans. They are also used to test hypotheses regarding human travel behavior, transportation and land use interactions, and the effectiveness of alternative investments or pricing policies [19, 20, 22, 28]. Transportation network simulation models are used to evaluate changes in economic activity or transportation system characteristics at a more disaggregate level. Traffic management policies, congestion reduction strategies, impacts of new development (such as housing or commercial centers) are some examples of network simulation model applications [14, 2, 25, 26]. More powerful and efficient computing has made possible wider application of micro-simulation models. These models are data intensive, requiring extensive and detailed information on vehicle flows.

Due to their recent rapid increase, freight flows are of growing interest within metropolitan areas. As the impact of commodity flows has increased, government planners and system operators have a greater demand for commodity flow information and for better methods to track, analyze, and monitor these flows as they impact transportation networks and nodes. Demand for better information and analysis tools is particularly strong at the metropolitan level, because access to disaggregate data is limited and analysis tools are not yet well developed [12, 23, 10].

The lack of data on truck traffic is particularly problematic. We have surprisingly little information on the characteristics of truck traffic, including its distribution across space and time. State highway transportation departments have "weigh-in-motion" (WIM) stations at key locations on the interstate highway systems that provide truck traffic data, but there are relatively few such stations (there are only 84 stations in all of California), mainly due to their high installation and maintenance costs. Additional truck data is obtained from special double inductive loop sensors, but these also are relatively scarce. Truck data on the arterial system is almost non-existent, because there is no easy way to collect such data.

For research and planning, as well as for some types of system monitoring, sample data are sufficient. Ideally, we would have accurate, low-cost methods for short-term data collection. In this project we developed components of a network of traffic sensors (NOTS). These are small, low-cost computer nodes, each with a portable inductive loop sensor. Each locally classifies a vehicle, and then adjacent sensors compare their values to select the most accurate classification.

In addition to research, planning, and system monitoring needs, there are many other potential applications that would benefit from a rapidly deployable data collection system. For example, SB 2650 imposed fines on terminal operators if trucks entering the terminal idled in queue for more than 30 minutes under certain conditions [11]. This law was intended to reduce diesel emissions. At the ports of Los Angeles and Long Beach,the law was enforced by a roving officer who estimated idle time based on his assessment of queue length. Because of the size and extent of the port complex, one roving officer had limited opportunity to systematically observe queues at terminal gates. A NOTS system that could be deployed on a random basis would have allowed for frequent and much more accurate observations, and hence more effective enforcement.

A second example is monitoring truck diversion from congested facilities such as the I-710, or truck diversion to avoid vehicle inspection. Local residents are increasingly concerned about truck traffic. It would be helpful to understand the patterns and extent of truck diversion in response to congestion in order to develop better estimates of the costs of congestion. A third example is measuring transaction times, such as container movements at ports or truck movements at large warehouse facilities. The key problem of sampling relevant to our proposal is that current static traffic management systems (TMS) do not provide sufficient flexibility for occasional sampling needed for research and planning for goods movement (topic area 1).

Although data collection is the primary motivation for our work, other applications include use of networks of traffic sensors (NOTS) around construction areas to manage traffic and improve worker safety (topic area 4), and to assist in traffic flow in emergency situations such as during a major evacuation (topic area 4).

1.2 Limitations of Current Approaches

Our research plans to address the lack of flexibility in current, static TMS by developing new, redeployable, networked vehicle monitoring systems. We have previously reviewed the limitations of current portable vehicle traffic monitoring systems [15]. We briefly summarize those limitations here.

Permanent monitoring systems: Extensive research exists on improving the performance of *permanent* monitoring systems, including better performing inductive loops, improving signature analysis, supplementing loop data with video sensing, and developing better sensors. We previously reviewed this work ([15], Section 2.1). The fundamental limitation of these systems is that, because they require permanently emplaced sensors, they cannot be easily be deployed around sites that need short-term data collection. Yet data collection for research and traffic modeling, planning for the traffic impact of local construction, or study of short-term phenomena (sporting events or traffic diversions due to construction) all require *short-term* data collection.

Deployable systems: The state-of-the-art in short-term deployable traffic monitoring is far less developed. While individual sensors such as tubes or tape-down inductive loops are relatively easy to deploy, an entire system is required to collect data.

Some semi-mobile traffic monitoring systems exist: A field operational test was conducted using a mobile surveillance and wireless communication system. The system was installed on a trailer that could be placed at roadsides. The core of the system was video image processing (VIP), used for vehicle detection, traffic volume, speed, and occupancy. Wireless radio allowed communications with the local TMC. The system was powered by a propane-powered generator [17]. A portable TMC was developed by the Minnesota DOT as part of a Smart Work Zone project. The system is mounted on portable skids and includes vehicle detection and surveillance, traffic control, driver information (via changeable message signs) and communications. Although relevant, these systems are orders of magnitude larger and more expensive than the approaches envisioned in our research and so they do not allow widespread use.

2 Literature Review

A careful review of current work was a major focus of the SURE-SE project. Chapter 2 of the SURE-SE final report considers current work on novel sensors, approaches to vehicle classification, and approaches to reidentification [15]. (We briefly summarize this above in Section 1.2.)

The focus of SURE-FT is to evaluate the use of multiple small sensors to collaboratively classify vehicles. We therefore review related work in sensor networks below, considering prior work in vehicle tracking, classification, and sensor fusion.

2.1 Sensing for Vehicle Tracking in Constrained Environments

In-road traffic sensors are ubiquitous in most urban environments. The need for efficient traffic flow has sparked significant investment in novel uses of both existing and new sensors. A number of sensor technologies have been considered. Pneumatic tubes and piezoelectric sensors detect wheel crossings; inductive loops and magnetic sensors detect vehicle mass; and infrared, ultrasound, radar or laser ranging, and video, employ different levels of imaging. We survey these elsewhere [15]. Important differentiators here are ease of deployment, robustness, and cost. In-roadway inductive loops are widely used and quite robust, but require construction to deploy. Video approaches remain relatively expensive, both in use and deployment (which may require elevation).

In spite of the large amount of research done on traffic sensors, systems for temporary deployment often fall back on simple pneumatic tubes, or manual, human observers, either physically present or interpreting videotape. For our work we use portable inductive loops because they retain the deployability of tubes but can provide much greater information.

Closest to our work is that of Oh *et al.* [21], Sun *et al.* [24], and Cheung *et al.* [4, 5]. Oh *et al.* use the same IST Blade sensor we do, but targeted at arterial speeds (20–45mph) rather than slow speed. In addition, they select a very different set of features, use a neural network for vehicle re-identification rather than classification, and so do not explicitly estimate vehicle speed or length. Sun *et al.* add a neural network to loop sensor output, explicitly trying to differentiate a custom set of categories including cars separate from SUVs, as well as larger vehicles. They report good accuracy, 82–87%, although they do not indicate if their errors are in the difficult-to-distinguish categories or not. Cheung *et al.* instead use custom sensor nodes with magnetometers, measuring the change in the Earth's magnetic field. They use both length-based classification and a novel "Hill Pattern Classification". While they obtain highly accurate vehicle counts (98%), their classification accuracy (82% into 5 FHWA classes corresponding to different axle-count large trucks) [5] is slightly better than ours (74% into 3 FHWA classes corresponding to passenger cars, small trucks, and large trucks). However, it is important to note that their classes are quite distinct and can be distinguished by axles counts and large differences in the length. We instead focus on FHWA classes 2 and 3, difficult to distinguish cars vs. small trucks We also explore the use of multiple features in a single signal, as well as multiple independent sensors to improve classification accuracy. Knaian use a small sized wireless node to count vehicles as well as monitor road conditions with embeded temperature and moisture sensors [18]. But it lacks the classification feature, which is the most difficult problem.

2.2 Sensor Networks for Vehicle Tracking in Unconstrained Environments

Vehicle tracking was one of the first problems for distributed sensor networks [3, 13, 29]. In general, these approaches focus on tracking relatively sparse (clearly distinct) targets without assumptions about target motion. To cope with these challenges they exploit multiple sensors for data from different viewpoints and use information theoretic techniques to estimate the vehicle path [3, 29]. More recent work has focused on less powerful nodes [7] and approaches to accommodate individual sensor noise [13], but still addresses relatively sparse targets. By contrast, our work focuses on densely packed vehicles on a busy roadway, and we exploit the capabilities of powerful cross-road sensors to make the problem tractable.

2.3 Sensor Fusion for Improved Accuracy

Sensor fusion is an important approach to exploit multiple sensors. Zhao *et al.* use information theoretic techniques to coordinate cooperation in a multi-sensor environment, selecting the sensor based on maximum information gain [29]. Brooks *et al.* explore collaborative signal processing and identify the level of sensor independence (how correlated or uncorrelated each is with others) as an important issue [3]. Gu *et al.* exploit cluster-based processing to correlate readings from multiple sensors [13].

3 System Description

Our long-term goal is to develop sensor networks that allow automatic accurate vehicle classification and re-identification. This section describes the hardware we propose to deploy to accomplish this task. In the next section we describe the algorithms we have developed for classification and to support deployment.

3.1 System Components

Our focus is a sensor network that is rapidly deployable, low cost, and sufficiently accurate. Our basic sensor network consists of several individual sensor nodes, each connected to a IST-222 high-speed detector card and a pair of Blade inductive loop sensors [16]. We selected the IST detector card because of its potential for sampling at high resolutions (up to 1.2 kHz), and the Blade inductive loop because of its sensitivity to vehicle features. The IST detector can be integrated with sensor network platforms such as the Intel Stargate [6] via USB, and we expect to use networks such as 802.15.4 for low-power, short-range communication. The

Figure 1: Deployment of a narrow inductive loop for a blade sensor.

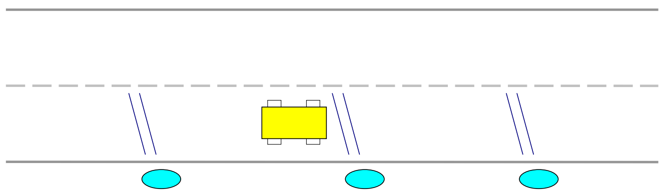


Figure 2: Deployment of three sensor nodes and six sensors along a roadway.

systems as a whole should be relatively inexpensive (currently less than US$4000). The complete system is easily deployable since both the sensor nodes and the Blade sensor can operate on battery power. Pairs of inductive loops must be taped down across traffic lanes using asphalt tape, thus requiring a brief interruption of traffic. Figure 1 shows deployment of a pair of loops as part of our trial. Figure 2 shows the logical configuration of our array of traffic sensors: each individual sensor node connects two closely separated blade sensors (a car is shown approaching the middle pair of sensors).

We report here on analysis conducted with data from two locations. First, we discuss our single sensor algorithms using data from one location. We then address multi-sensor fusion using data from two locations. We have not yet integrated our multi-sensor algorithms with communications protocols; currently we manually process multi-sensor data at a central site. We believe a distributed implementation is not difficult, and plan to exploit sensor deployment information to make the system largely self-configuring, as in prior work [27].

3.2 System Integration

Most of the evaluation described in this report was carried out with off-line, post-facto analysis of stored traces. This approach has been essential to refine and improve our algorithms with a consistent dataset. However, we are working towards developing an on-line system capable of real-time vehicle classification.

To this end we began integration of our software into a package suitable for on-line operation. We have completed integration of all single-node processing, including data collection, segmentation, and single-node classification. It also includes our new autoconfiguration algorithms described in Section 4.3. This software is freely available from the our project website `http://www.isi.edu/ilense/sure/index.html`.

Although we planned to use embedded computers running Linux as our sensor nodes, device drivers

action	**time required**
physical deployment	30 minutes
probe vehicle collection	10 minutes
auto-configuration execution	$<$ 1 minute

Table 1: Approximate deployment times of integrated system.

for the IST blade sensors are currently only available for Windows. Our current system therefore runs on Windows laptops. IST informs us that they are working on Linux drivers, and our code should easily port to an embedded platform when drivers are available.

We tested both classification and auto-configuration algorithm with a short experiment conducted on July 28, 2006 at the roof of ISI parking structure as described in Section 5.3 We collected 36 vehicle signatures of three different vehicles. This experiment confirmed that automatic deployment was effective and that our integrated algorithms work well.

Deployment required several steps: first we physically deployed the sensors. We then ran our probe vehicle across the sensors. We run the probe vehicle over the sensor one or two times to run our auto-configuration algorithms. Approximate deployment times of each phase are shown in Table 1. Physical deployment of the a pair of sensors required 30 minutes, during which time the roadway must be closed or partially closed. We expect that could be reduced with more deployment experience. Additional configuration (primarily probe vehicle data collection) can be done after the roadway is opened (although with a mix of vehicles, the operator must currently manually select which signatures are of the probe vehicle).

In addition, we confirmed that our integrated system worked well. All 36 vehicle crossings resulted in positive detections and accurate classification in real time.

Although promising, this experiment is quite preliminary. We must complete integration of communication *between* nodes and support for multi-node collaboration. In addition, this experiment with the integrated software involved only three vehicles and was done in a parking space instead of a real road. Additional experience, including deployment on real-roadways and with a wider selection of vehicles is needed to more fully evaluate the system.

3.3 System Availability

The software we developed as part of our system and to analyze our results is available on request from the authors.

4 Algorithms for Vehicle Classification

The SURE-FT project has revised the stand-alone classification algorithms which is described in previous SURE-SE project and improve the accuracy with collaborative sensor networks. We also develop an automated configuration to achieve easy and quick deployment. This section describes these classification algorithms (Section 4.1), the types of error we expect and how we recover (Section 4.2.1), and how we automate system configuration (Section 4.3).

4.1 Classification Algorithms

We first review our basic classification algorithms, with single sensors and wheelbase as the classification feature. We then consider the benefits of multiple sensors.

4.1.1 Single-Sensor Classification

Signal processing at an individual sensor is fairly traditional: we begin with noise removal, segment the data into individual vehicles, extract features pertinent to our analysis, then classify vehicles; we examine each of these steps next.

Noise elimination: In some cases we have observed significant crosstalk, environmental noise, and signal drift in our measurements. In addition, communications from the IST card to the host computer is not perfect. Crosstalk arises when inductive loops driven by different sensor cards are close to each other, and also due to slightly different sensor clock rates in the case that multiple sensor cards are attached to the same sensor node.

Drift and noise occur due to temperature change and other environmental effects. We filter each of these out using standard techniques. We observe around a 14% data loss on the USB bus between the sensor and the host (this behavior is a known problem of the specific model of detector cards we were using); we correct for this by interpolating the missing data.

Segmentation: When noise is eliminated, we are left with a continuous signal. We next isolate individual vehicles with a three-step process. We first detect active segments by observing large signal deviations from a running mean. We then merge temporally close active segments to allow for vehicles that have "flat" areas between wheel wells. Finally, we grow segments by half their length at front and back to ensure we capture a complete segment, including leading and trailing features. As a special case, when growing a segment would cause overlap with a neighboring segment, we grow to the midpoint between segments. Ideally, after segmentation, each segment corresponds to exactly one vehicle. In practice we find that occasionally (about 5% of the time, see segmentation errors in Table 9) vehicles that are very near to each other appear in a single segment.

Feature extraction: We experimented with several possible features for vehicle classification, including axle count, body width, and wheelbase (axle-to-axle distance). We converged on a two-level set of features. We directly extract the edges of wheels (70% wheels points, described below), then figure these and estimate speed and wheelbase. Our first goal is to determine wheel edges. Figure 3 shows a sample signature of a two-axle car crossing a sensor. Wheels show up as large dips in the signature, the underbody as bumps between the wheels. Because the vehicle crosses the inductive loop at an angle, each wheel of the same axle produces a distinct dip located near the other. We experimented with different algorithms to reliably extract each wheel. Our two main approaches were to identify peaks and to identify large changes in direction. Although peak identification is attractive, consistent results are difficult because peaks tend to be rounded, particularly at higher sampling frequencies (because wheels are round). In addition, depending on the angle the car crosses the sensor, we may get two clearly distinct peaks or a single merged peak. Because of these difficulties we adopted the "steep slope" algorithm shown in Algorithm 1.

It is important that our algorithm adapt to a wider range of vehicle speeds. To do so, we adjust the parameter v based on vehicle speeds. First, we start with a default v value 240 ($0.2R$, where R is the sampling rate) to estimate the vehicle speed, since it is not necessary to separate each wheel well to get speed estimates. Then, we adjust v according to the estimated speed as shown Algorithm 1. If v is too large, we cannot separate the wheel wells; on the other hand, if it is too small, we cannot capture the all wheels in the wheel well. In case of extremely low speed (less than 5mph), we use 680 ($0.4R$) for v to prevent from collapsing every wheel well into one. The other parameters are somewhat arbitrary. We initially chose the S value as 70%, with the goal of identifying a point on the steep part of the signature. We studied values from 30–80% and found they did not affect classification accuracy, as shown in Figure 4.

Speed Estimation: To estimate vehicle speed we compare the time difference when a wheel-point crosses each of the two closely placed loops. Figure 5 shows a time-correlated sample from both of the paired loops for one signature. We compare pairs of shapes (squares or circles), and similar observation at the front and back of the first and last wheel wells. Since the loops are a known distance apart, a speed estimate is simply

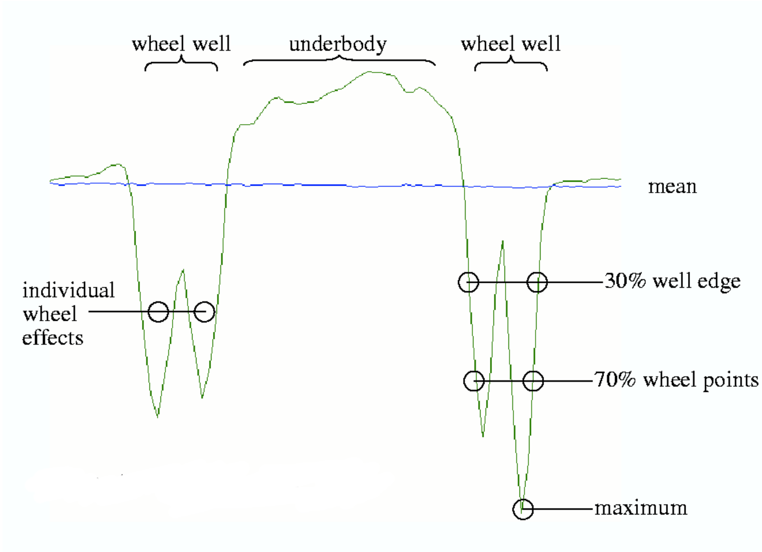


Figure 3: Sample signature indicating vehicle parts (signature #280, site BN)

Algorithm 1 The steep-slope algorithm for wheel edge extraction.

Normalize the signature values from 0.0 (lowest underbody) to 1.0 (highest wheel peak)
Compute m, the mean of all sensor readings over entire signature
Identify wheel wells by finding the first value greater than $m+(L*m)$ through the last value greater than $m+(L*m)$, allowing up to v consecutive values below $m+(L*m)$
For the first and last wheel wells:
  Find the maximum value M in the well
  Define the start-wheel-point as the first value in the well greater than $S*(M-m)$
  Define the end-wheel-point as the last value in the well greater than $S*(M-m)$
Parameters:
$v : 2/(\mathit{VehicleSpeed}) * \mathit{SamplingRate}$, number of samples allowed below $m+(L*m)$ in a wheel well (between wheel peaks)
$L : 0.3$, wheel well start threshold (fraction of mean)
$S : 0.7$, target for steep slope (fraction between mean and peak)

the distances divided by the time between the same feature at each adjacent loop. To reduce error, we match the wheel-points for the start and end of the first and last wheel wells, giving us four estimates of speed. We then average these values. We discuss how this approach addresses different errors in Section 4.2.2, and quantify these benefits in Section 6.2.1.

Wheelbase: To estimate wheelbase, we observe the front of the first and last wheel wells: solid shapes (squares or circles) in Figure 5. With two paired sensors, we can get four estimates front and back of sensors 1 and 2.

Classification: Given a speed estimate, we classify vehicles by wheelbase length, the distance from the first to the last wheel axle. Given our wheel-points on the first and last wheel wells, we have four wheelbase estimates on each of our two paired sensors. As with speed estimates, we average these four readings.

Finally, we map length to vehicle classification as described in the next section.

Clearly our classification algorithm is quite simple. A much more sophisticated system would, for example, match the entire signature against a database of known vehicle types. However, even this simple

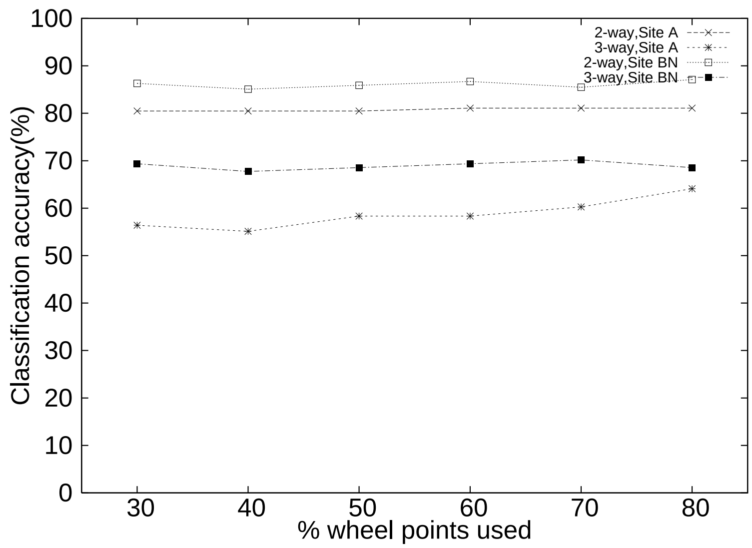


Figure 4: Classification accuracy as a function of wheel point percentage.

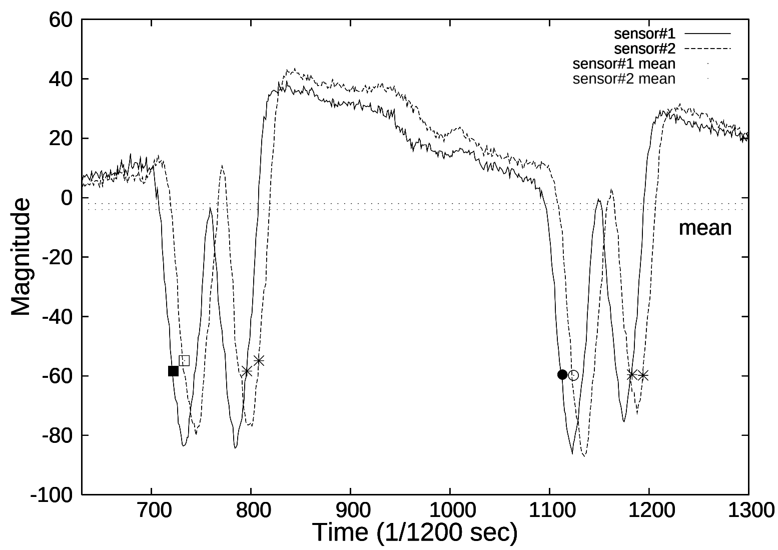


Figure 5: A signature with paired sensors (signature #130, site BN)

classification system is sufficient to explore the use of multiple sensors to reduce error. In Section 7 we describe future directions defining more appropriate classification schemes.

4.1.2 Classification by Wheelbase

Table 2 shows our mapping from length to our two- and three-category classification systems and the relationship to the FHWA classes. We selected the two-category system because automatic classification of truck vs. non-trucks is relatively easy, while we will show that distinguishing cars from SUVs (P from S*) is much more difficult and thus represents a “worst case” for automated classification systems.

A critical problem with the FHWA system is that the boundaries between classes 2 and 3 are indistinct. In fact, the FHWA website says “because automatic vehicle classifiers have difficulty distinguishing class 3 from class 2, these two classes may be combined into class 2” [8]. This problem applies also to human

Table 2: Length-based vehicle classification

FHWA classes	meaning	symbol	wheelbase length
Two-category system			
2 and 3	non-trucks	non-T	< 170”
4 to 13	commercial trucks	T	≥ 170”
Three-category system			
2	passenger cars	P	< 118”
3	small trucks/SUVs	S*	118 to 170”
4 to 13	commercial trucks	T	≥ 170”

Table 3: Two-category classification with perfect sensors

Class	Total	Correctly classified	Incorrectly classified
non-T	42	41	1
T	5	4	1
Total	47 (100%)	45 (96%)	2 (4%)

Table 4: Three-category classification with perfect sensors

Class	Total	Correctly classified	Incorrectly classified
P	24	24	0
S*	18	12	6
T	5	4	1
Total	47 (100%)	40 (85%)	7 (15%)

observations (we quantify human accuracy in the SURE-SE); we next consider how length relates to classes.

To evaluate the wheelbase-based classification with perfect sensors we surveyed wheelbase lengths of 47 vehicles from Ford [9] and government sources [1]. Tables 3 and 4 show classification accuracies based on wheelbase assuming perfect length determination. Even though we classify vehicles with their exact lengths, not all surveyed vehicles are correctly classified: 96% of vehicle models are correctly classified in two-category classification. In three-category classification only 85% of models are correctly classified, since there are many SUVs on either side of our threshold, regardless of where it is placed.

These results assume static analysis, in that the percentages are based on numbers of vehicle types. In practice, the population of vehicles of each type varies, as does the particular set of vehicles observed at any site. Therefore, dynamic measurements may differ depending on the mix of observations.

4.1.3 Using Multiple Sensors

A defining characteristic of sensor networks is the use of many relatively inexpensive sensors. We therefore wish to explore if multiple sensors can improve the best-possible classification results of a single sensor. Our hypothesis is that classes of errors are independent, so combining values from moderately separated sensors can eliminate these errors.

As shown in Figure 2, we place several pairs of sensors at several places on a roadway. We expect sensors to communicate through a local, low-power, wireless network such as provided by 802.15.4 or similar networks. Sensor nodes will not share raw sensor readings, but instead individual evaluations of vehicle type,

Table 5: Types of errors in vehicle classification

Type of error	Generality	Dependence	Addressed	Observed
Environmental noise	General	Either	(In-sensor) or post-facto	Yes
Sensor failure	General	Independent	(Multi-sensor)	No
Insufficient sampling	General	Independent	(Design)	Yes
Vehicle type	Application	Dependent	(Multi-sensor)	No
Mis-channelization	Application	Independent	Multi-sensor	Yes
Imprecise speed	Application	Dependent	Single sensor	Yes
Changing speeds	Application	Independent	Single sensor	Yes
Mis-segmentation	Application	Independent	Multi-sensor	Yes

coupled with data about their confidence in the classification. Such a system must have several components: a configuration system to automate initial deployment, communications protocols to share information between sensors, a signature matching algorithm to identify which signatures at one sensor correspond to signatures at another sensor, and a classification preference algorithm to select which classification is best. We plan to exploit the simple, constrained topology of the road, so configuration and communications are straightforward as each sensor interacts with its immediate neighbors. Signature matching and classification preference are the keys to improving accuracy with multiple sensors. We discuss these algorithms in Section 4.2.3 after reviewing potential types of error.

4.2 Types of Errors and Error Recovery

To consider how multiple sensors might improve accuracy, we first evaluate the types of error that arise in this application, then consider how to make a single sensor as effective as possible, and finally how multiple sensors can further improve accuracy.

4.2.1 Types of Error

We review the types of errors we expect in Table 5. There we evaluate each error for its generality, if it is specific to this application or applies to all sensors; dependence, if we expect multiple sensors to exhibit this error consistently or independently; how we address it, in-sensor with multiple estimates or multiple sensors; and if we observed it in our examples.

We observed significant amounts of *environmental noise* in our data, both due to temperature drift and sensor cross talk. A later revision of the IST Blade sensor handles noise elimination in the sensor itself, but for our data collection experiment we filtered noise manually post-facto. Inductive loops respond to vehicle mass relative to its distance from the sensor, thus they are less sensitive to vehicles that are higher off the ground. Loop sensitivity can be controlled by adjusting width, so potentially multiple loops of different widths could detect a wide range of vehicles. For our main experiments we used a loop width of about 4 inches.

We did not observe any *sensor failure* in our system, but it would be an issue for larger deployments.

An *insufficient sampling rate* or too close placement of sensors can result in imprecise speed and length estimates, since a change of a single sample interval corresponds to a noticeable change in estimate. Sampling rate or sensor distance must be adjusted to expected speeds, as we explore in Section 6.1.

Vehicle type errors, refer to different distances of vehicles from the ground. We did observe distance affecting the quality of sensor signatures, however it was not a major cause of misclassification.

Mis-channelization is when only part of the vehicle crosses the loop because it is changing lanes. *Changing speeds* occur when a vehicle alters its speed over the sensor, making estimation difficult. *Mis-*

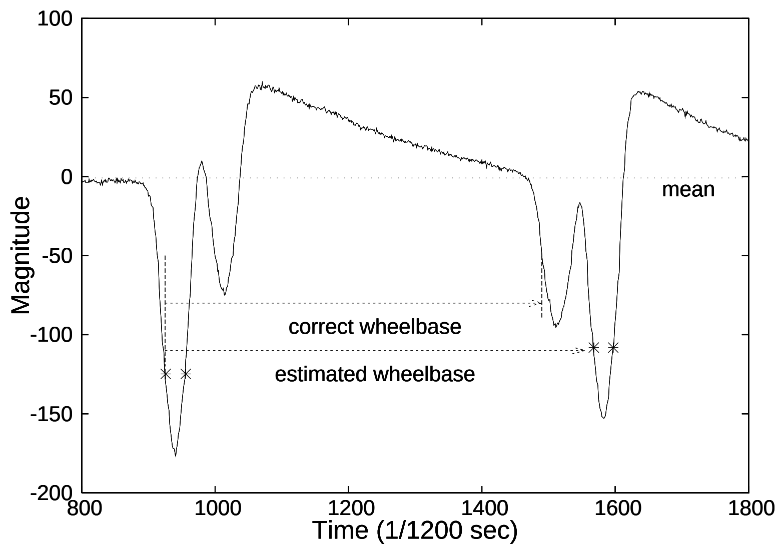


Figure 6: A signature with missing wheels (signature #96, site BN)

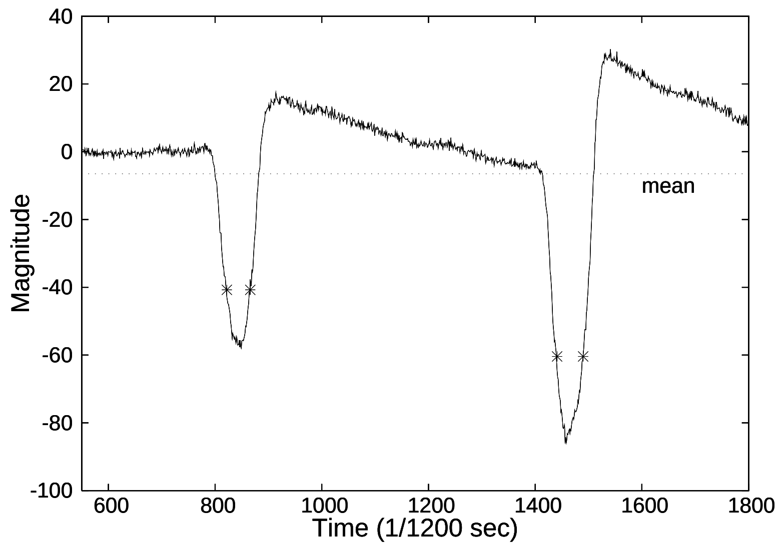


Figure 7: A signature with channelization error (signature #251, site BN)

segmentation occurs when two cars travel so close that they appear to the sensor to be a single vehicle. All of these errors are specific to vehicle classification, but each occurs independently at different sensors and so should be correctable in the sensor network. We observe and correct several mis-segmentation errors as described in Section 6.2.1.

4.2.2 Single-Sensor Error Recovery

We used three general techniques to improve individual sensor readings: sharp feature detection, internal consistency checking, and cross-checking with multiple features .

Our early approach identified speed and wheelbase by using peaks of the signature to estimate the exact wheel locations. This approach proved inaccurate at high sampling rates because wheels provide rounded features to the inductive loop, since the loop has an effective low-pass filter, and wheels are round. To address these problems, we shifted to identifying the sharp slopes that correspond to the edges of the wheel peaks, as determined by the S threshold, a fraction of the distance from mean to the peak value (currently 70%). Since the slopes on the edges change rapidly, slopes tolerate much more error than peaks. In fact, we experimented with thresholds corresponding to 30–80% of the distance between the mean and maximum sensor values and found classification accuracy unchanged.

Second, we check the features for internal consistency. Our primary approach is to evaluate where detected wheels are placed inside the wheel well. We expect to get two peaks in each wheel well, corresponding to the left and right wheels. However, if one of the peaks is much smaller than the other, sometimes it is missed by the steep-slope algorithm. For example, in Figure 6, if we miss a right wheel in the first and the left wheel in the last, the estimate will be longer than the proper estimate. One wheel may be omitted from a wheel well when channelization errors occur (Section 4.2.1). Ignoring these errors can result in significant inaccuracy in wheelbase estimation. Figure 7 shows a signature in channelization error.

To solve this problem, we take several steps to examine wheel placement for internal consistency. (We explored adjusting the steep slope target S to catch low peaks, but in general one cannot catch all low peaks and be robust to noise.) We count the number of wheels in each wheel well. If we believe a wheel is missing, we identify which wheel is present, or perhaps determine that we cannot tell which wheel we observe. There are four possible wheel placement states: both wheels present, left wheel or right wheel missing, and unable to determine which wheel is missing. There are 16 possible combinations of these cases when we consider detection at the first and last wheel wells.

When both wheels are present we get two wheelbase estimates per loop. If a wheel is missing but we can identify which wheel is present we calculate a single wheelbase estimate with the present wheel. However, when we cannot determine which wheel is present, or if different wheels are present in the front and back wells, we know that our wheelbase estimate will be incorrect. In these cases we report our best estimate along with a lower confidence value in this estimate, potentially allowing multi-sensor error recovery to select a better estimate at another sensor as explained in the next Section 4.2.3.

Finally, we obtain multiple estimates of each feature in a single signature. We can find four wheel-points for each signature in the beginning and ending of the front and back wheel wells. (These are indicated by stars in Figure 5, in addition to the squares and circles.) This gives us four estimates of speed and length. Although these estimates are not completely independent, averaging them provides much less variance than any individual reading. This approach also partially corrects for vehicles that change speed over the sensor. We quantify the benefit of this in Section 6.2.1. Although reduction in variance does not necessarily translate into improved accuracy, it does imply less susceptibility to noise. We quantify the reduction in variance in Section 6.2.2.

4.2.3 Multi-Sensor Error Recovery

To use multiple sensors to reduce errors we must identify when multiple signatures correspond to the same vehicle and then which classification is most accurate: signature matching and classification preference.

Signature matching: We plan to exploit the constrained topology of a roadway to simplify signature matching. If sensors are placed on a roadway without intersections or exits, then the order of vehicles and signatures is fixed and so signatures can be matched based on timing and ordering. Missing signatures can be inferred by gaps, and mis-segmentations by a very long signature at one sensor followed by a short signature at the next.

Classification preference: Given two matched signatures with different classifications one must choose which classification is more likely to be correct. We are experimenting with two algorithms: quality-best and shortest-best.

The *quality-best* algorithm favors sensors that are able to consider multiple estimates that report consistent values. This approach addresses speed variability, since a larger number of speed and length estimates reduce the impact of variability and allow the sensor to estimate its confidence in the estimated speed and length (more variance indicates less confidence in the estimate). We also adjust the confidence according to the internal consistency of the signature by considering where we believe wheels are placed in a signature. If there are missing wheels and their placements are inconsistent, we reduce the confidence value to half of what we get from the variance.

The *shortest-best* algorithm is much simpler. In our experiment we found some errors were due to mis-segmentation. We can detect mis-segmentation as a long signature at one sensor with two short signatures at the other. Our system is more likely to merge two adjacent signatures than split a long signature, therefore in these cases the shortest-best algorithm selects the two-signature interpretation over the single long signature.

Finally, for analytic purposes, we consider an *oracle* algorithm. The oracle algorithm assumes a perfect classification preference algorithm that always chooses the correct classification if it is present at either sensor. Such an algorithm is impossible to realize in a real system—we can implement it only because we already have ground truth. We present it to provide an upper bound on how well sensor fusion can do.

Current Status: We have implemented classification preference based on sensor quality, and multi-sensor comparison. We have not implemented a complete matching algorithm, since in our data collection experiment sensors were separated by a large distance and an intersection, so for our preliminary analysis we manually associated signatures at two sensors.

4.3 Automatic Configuration

A goal of our system is that it be easily and rapidly deployable. This goal motivated our choice of hardware platforms (small, wireless devices) and sensors (inductive loops that can be simply taped down on the roadway). However, in addition to physically deploying the system, the software requires configuration as well.

We therefore designed an *automatic configuration* system to easily determine any software parameters required for system operation. Automatic configuration involves driving a *probe vehicle* over the sensor multiple times. The probe vehicle can be any vehicle; in our experiments we used a small passenger car.

The system collects data to determine *segmentation thresholds* needed to determine where signatures begin or end. We also automatically determine the *sensor separation distance*, a value needed to compute vehicle speed. We describe each of these below.

4.3.1 Segmentation Thresholds

Segmentation is an early step in classification where raw sensor data is split into per-vehicle segments (see Section 4.1.1 for a complete description of the algorithm). Segmentation requires a threshold to separate the active vehicle signal from background noise.

Ideally we could assign a fixed segmentation threshold based on laboratory data. Unfortunately, we found that the threshold used varied based on the version of sensor detect card, and the loop construction (type of wire, number of turns, etc.). While we manually configured it for our preliminary experiments, we do not expect end-users to hand configure it and therefore designed an automatic procedure.

Algorithm 2 shows the algorithm we developed. We use a known number of probe vehicle detections as "ground truth", then iteratively adjust an initial value until we get a matching number of detections from the sensor. The initial threshold factor is decided by the heuristics from our July 2006 rooftop experiment (Section 5.3).

We report on this algorithms effectiveness in Section 6.3.

4.3.2 Sensor Separation Distance

Knowing the exact distance between loops is crucial to get the proper speed and length estimation, since we use a pair of loops to calculate the speed of a vehicle as described in Section 4.1.1. Therefore every pair of

Algorithm 2 Automatic segmentation threshold

Observe signal of the probe vehicle.
  Record the stream of signal during this period without any segmentation.
  While observing the stream, calculate windowed standard deviation, E (signal energy).
  After the probe vehicle runs over the sensor N_{actual} times, find $Max(E)$.
  Set the temporary thresholds
    $Th_{Hi} = T * Max(E)$
    $Th_{Lo} = 0.5 * Th_{Hi})$
Run the segmentation algorithm with above thresholds on the recorded signal.
  Initialize the threshold adjustment size
    IF $N_{signature} < N_{actual}$ THEN $\Delta = \Delta_{inc}$
    IF $N_{signature} > N_{actual}$ THEN $\Delta = \Delta_{dec}$
  IF $N_{signature} == N_{actual}$ THEN set the thresholds for the classification
    ELSE adjust the thresholds:
      IF $N_{signature} < N_{actual}$ THEN increases thresholds by Δ
      IF $N_{signature} > N_{actual}$ THEN decrease thresholds by Δ
      Update $\Delta = \Delta/2$
  Iterate above steps until matches to actual count.
  If it doesn't match within two iterations, then declare "Failed to find thresholds"
Parameters:
T : Initial threshold factor, 0.25
$N_{signature}$: Number of signatures
N_{actual} : Actual vehicle count
Δ_{inc} : initial increment size of threshold $((1-T)*Max(E))/2$
Δ_{dec} : initial decrement size of threshold $(T*Max(E))/2$

loops should be measured exactly or they should be separated by precise distance when deploying the loops on the road. Either way could require a long interruption of traffic. The experiment described in Section 6.1 shows that the distance between two loops affect the accuracies of speed estimation and length estimation. Hence, a small error in measuring sensor separation distance could result in consistent and significant error in vehicle classification.

To eliminate this measurement error we calibrate the sensor separation distance using the probe vehicle length. This procedure inverts our regular algorithm: given a known vehicle length and a measured time to cross two loops, we back-compute the physical distance between loops. This computation is possible because we know the exact length of the probe vehicle. In addition, we can use multiple readings of the probe vehicle to correct for measurement error.

Auto-configuration of sensor separation give two benefits: First, it simplifies deployment, since sensor distance is not critical. While loops must be parallel to each other, their separation can be arbitrary, since we measure it and correct for variation in a given deployment. Secondly, when there are multiple pairs of sensors, each may be deployed with different separation distance, yet this variation can be correct for by auto-configuration. Finally, if the system is to operate for long durations (more than a few days), potentially we can easily re-calibrate the sensor separation to account for any movement due to vehicle traffic. (We have not yet tested this case.)

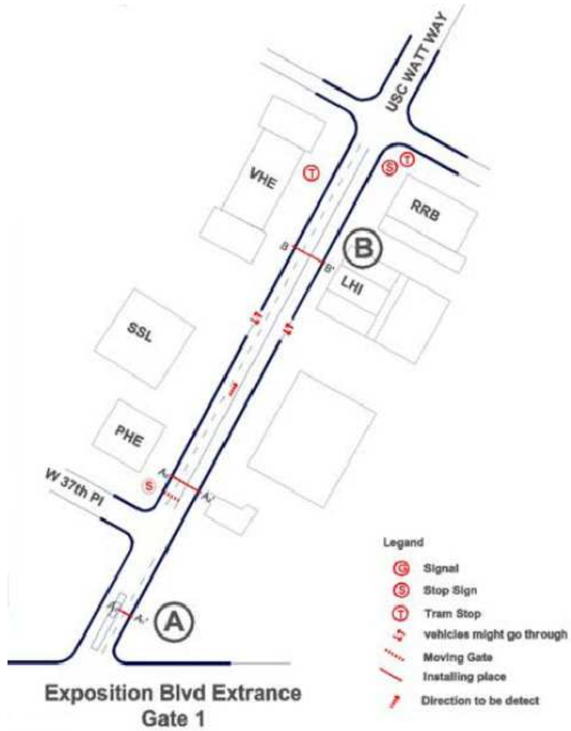


Figure 8: Placement of sensors for data collection.

4.3.3 Other kinds of automatic configuration

The approaches described above show the principles of automatic configuration needed for the algorithms developed in this paper. Additional future automatic configuration may include sensor location and relative position of sensors. For example, we could automatically infer which sensors are deployed in which traffic lanes, the direction traffic flow, and which sensors are downsteam of each other.

5 Data Collection

In SURE-FT we used several datasets to draw our conclusions. Our primary dataset is a collection of 1500 vehicle signatures taken in August 2004 at USC as part of the SURE-SE project. To this we added two additional, small scale experiments taken on a parking garage rooftop at USC/ISI. We describe each of these below.

5.1 USC August 2004 Data Experiment

From 7 am to noon, August 6, 2004, we collected traffic data at the USC campus. Working with Steven Hilliard of IST, we collected 1500 detections of vehicles at three locations. Sensor data was supplemented with human observers and videotape to provide ground truth data. We selected three locations on internal campus streets to get a mix of low- and moderate-speed traffic. We selected a data collection day when construction was underway on campus, allowing us to capture a mix of periodic traffic, including the USC shuttle bus, construction traffic, including cement mixers and 18-wheel trucks, and general automobile traffic. In addition to general traffic, we selected two passenger cars and ran them over each sensor 10 or more times to provide a baseline known vehicle to evaluate re-identification and sensor consistency.

Details about exact deployment locations are available in a technical report [15]; here we report on two locations: site A, near the campus entrance, and site B northbound (or BN), about 100m north of site A, past an intersection. Figure 8 shows this layout. Site A is next to a parking kiosk and has two lanes. Typical vehicle speeds are quite slow (mean speed 8mph); stops in lane #1 (immediately next to the kiosk) were frequent. Here we report only data from lane #2, further from the kiosk. Site B consisted of north (BN) and southbound (BS) lanes, mid-block. Mean speeds were 16mph. For most of our analysis we consider site BN. We use site A to confirm our BN results and see how our approach works on slower traffic, and to examine multi-sensor fusion for vehicles that pass through both sites A and BN. We use the data from 9:15 am to 10:50 am, discarding earlier data because of incomplete video ground truth, and later data because of numerous records from our two test cars.

Our re-analysis of this data with updated algorithms appears in Section 6.2.

5.2 USC/ISI Rooftop December 2005 Experiment

To examine the accuracy of a single-sensor regarding the sensor separation and speed of vehicles, we conducted an experiment on the roof of ISI parking structure. We took data on December 7, 2005.

Our goals were several: to collect data to calibrate sensor speed detection against “ground truth” from a radar gun, to collect data to evaluate alternatives about sensor loop spacing, to verify tradeoffs in sensor spacing and sampling frequency.

The experiment involved two blade loops, an IST detector card driven off of a laptop computer, with a video camera to record activities, and a radar gun (Model: Bushnell 10-1911, accuracy: $\pm$1mph, precision: 1mph) to establish ground truth.

Our basic blade sensor loops were deployed with a 1.5 inch loop width, the loop wound one time. We looked at two separations of loop pairs: 18 inches and 36 inches. We used two test vehicles: *J* (a 2000 Toyota Celica GT) and *U* (a 2001 BMW 330ci).

Both targets were tested at 10 and 20mph , with each vehicle tested 10–20 times to collect data to evaluate repeatability. Results of this experiment are described in Section 6.1 Data from this experiment is on the 061206 CD-ROM in `TrafficSensor_v1/data/2005_12_07` directory.

5.3 USC/ISI Rooftop July 2006 Experiment

We tested both classification and auto-configuration algorithm with a short experiment conducted on July 27, 2006 at the roof of ISI parking structure. We collected 36 vehicle signatures of three different vehicles. We deployed blade sensor inductive loops constructed of 4 winds of 22 awg wire. Sensors were deployed at a 25 degree angle to traffic with an 18 inch separation between pairs of loops.

For these experiments we configured the IST card in automatic configuration. (The IST card has a dip-switch to change sensor sensor sensitivity according to attached loops types. When standardized, conventional loop are attached, it should use the preset sensitivity. Since we use custom loops, we chose the “automatic” mode so that it could determine the proper sensitivity based on the attached loops.) These experiments involved three vehicles: *U*, a 2001 BMW 330ci, wheelbase 107.5 inches; *J*, a 2000 Celica GT, wheelbase 102.5 inches; and *D*, a 2000 Acura TL 3.2, wheelbase 108.1 inches.

We used each vehicles as a probe vehicle (as described in Section 4.3), then collected 8–11 signatures of each vehicle. Results from this experiment are described in Section 3.2. Data from this experiment is on the 061206 CD-ROM in `TrafficSensor_v1/data/2005_07_27` directory.

Table 6: Single sensor accuracies for speed from the 18” and 36” sensor separation distance with 300Hz sampling. Standard deviations are given in brackets.

Sensor separation	Target speed (mph)	# samples	Achieved speed (mph)	Estimated speed (mph)	Absolute error (mph)	Relative error
18”	10	5	13.5 [1.97]	15.6 [1.36]	2.1	17.1%
	20	6	19.0 [1.87]	21.6 [2.94]	2.6	13.3%
	Overall	11	16.0 [3.40]	18.3 [3.73]	2.3	15.4%
36”	10	10	12.6 [0.84]	13.2 [1.54]	1.3	10.4%
	20	10	19.3 [0.94]	20.2 [0.69]	0.9	5.0%
	Overall	20	16.0 [3.55]	16.7 [3.77]	1.1	7.7%

Table 7: Single sensor accuracies for wheelbase from the 18” and 36” sensor separation distance with 300Hz sampling. Standard deviations are given in brackets.

Sensor separation	Actual length (inch)	Estimated length (inch)	Absolute error (inch)	Relative error
18”	107.5	113.4 [2.80]	6.0	5.6%
36”	107.5	107.7 [1.98]	2.6	1.6%

5.4 Survey of Vehicle Models and Wheelbases

Finally, to evaluate the effectiveness of perfect sensors, we surveyed 47 vehicle models of a single manufacturer (Ford) in comparison to the FHWA classification system. The results of this evaluation are found in Section 4.1.2.

5.5 Data Availability

Complete copies of the data described in this paper are available on request from the authors, with information available on the project website `http://www.isi.edu/ilense/sure/index.html`. This includes both the new experiments done as part of SURE-FT (available on one CD-ROM), and from the USC experiments done as part of SURE-SE (provided as one CD-ROM of data and several DVDs of video ground truth).

6 Experimental Evaluation

This section summarizes our new experimental results: verification of speed detection (Section 6.1); reanalysis of our USC dataset with our updated algorithms (Section 6.2), both for error handling at a single sensor, comparison to human counting, evaluation of the features that we chose, and use of multiple sensors.

6.1 Single-Sensor Calibration Experiment

To examine the accuracy of a single-sensor regarding the sensor separation and speed of vehicles, we conducted an experiment on the roof of ISI parking structure as described in Section 5.2. We tested two sensor separation distances: 18” and 36” at 300Hz sampling rate. We also experimented with two target speeds: 10 mph and 20 mph to demonstrate that different speeds affect estimation accuracy. Ground truth vehicle speeds are established by using a radar gun (Model: Bushnell 10-1911, accuracy: $\pm$1mph, precision: 1mph).

Table 6 and 7 shows detailed results. Speed estimation (Table 6) indicates that sensor readings are within 5–17% error. Linear regression of the correlation of estimated speed and radar gun speed shows an r-square of 0.85–0.90, indicating a very strong, but not perfect correlation between speed and ground truth speed. Although this accuracy is good, for speed we have a very poor reference "ground truth": the precision of the radar gun is only 1mph, so we believe some of this error corresponds to the poor precision of those measurements. In fact, with the wide-spaced sensors (36"), error is within this precision. With narrow-spaced sensors, errors are around 2.3mph, confirming that accuracy falls off with narrow-spaced sensors at these sampling rates. Finally, we observe that absolute error in speed is roughly constant for these speeds, and the relative error actually decreases.

When we turn to length (Table 7), we see much higher accuracies. The length estimation error is less than 6% which indicates that we can accurately estimate length using a single sensor. We also observe that the wide-spaced sensor is again more accurate. Note that length estimates are based on speed estimates, so we expect these results to be dependent. The main difference is that we have accurate ground truth of length.

We didn't investigate exhaustively the effect of distance and sampling rate on the accuracy, but we at least confirmed the configuration we used is appropriate for speed and length estimation. In Section 6.2, we experimented with 4.5" separation distance and 1200Hz sampling rate. We didn't explicitly examine this configuration, but the accuracy would be roughly equal to the configuration of 18" and 300Hz, because accuracy is proportional to the product of distance and sample rate.

6.2 Re-analysis of USC Data

We evaluate new algorithms with the data set collected at USC campus on August 6th, 2004. A description of our prior analysis of this dataset can be found in Chapters 3 and 4 of the SURE-SE final report [15] and summarized in Section 5.1.

6.2.1 Error Recovery in a Single Sensor

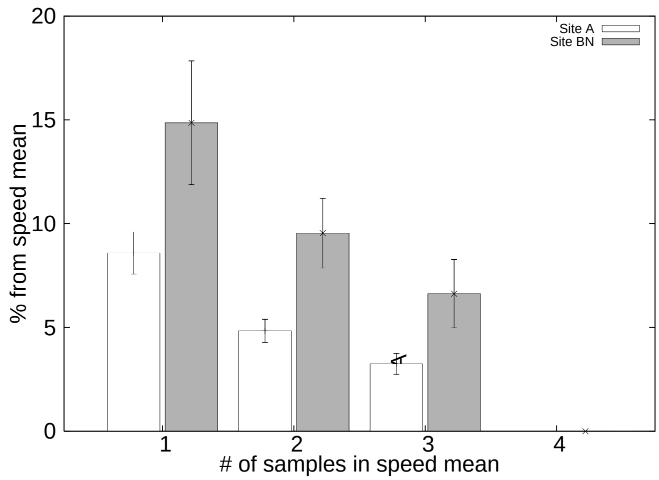
We first evaluate single-sensor techniques to improve estimation consistency.

To evaluate the benefits of multiple features we compare the amount of variance we see across multiple readings. In Section 4.1.1, we show that we can determine up to four estimates of speed and length from each sensor pair. We expect that two classes of error, imprecise speeds (due to sensor inaccuracy), and changing speed (if cars alter their speed over the sensor), can be addressed by exploiting multiple estimates at a single sensor.

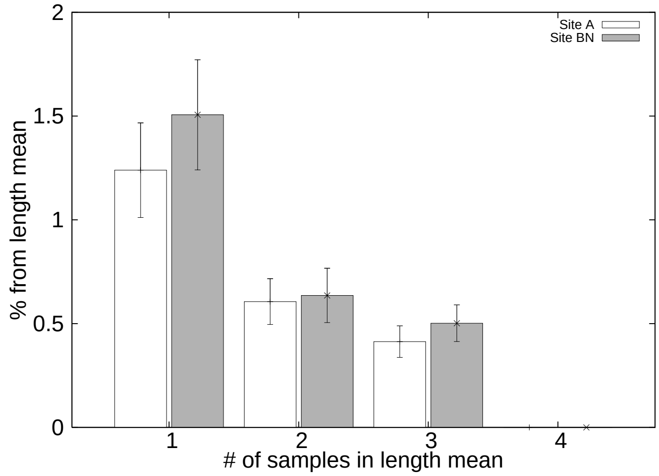
Figure 9 shows how variance changes when we consider 1, 2, 3, or all 4 estimates of each speed and length for data from both sites A and BN. Unfortunately we do not have ground truth about speed and length for our main experiment, so we computed our best-estimate of speed and length by taking the mean of all four measurements. For each signature, we compute the difference of an n-sensor estimate against this best estimate. We consider all possible combinations of n sensors for each signature, so the 1-estimate values consider 4 measurements per signature, the 2-estimate considers 6 combinations per signature (combinations of estimate 1-2, 1-3, 1-4, 2-3, 2-4, and 3-4), and the 3-estimate considers 4 per signature (estimates 1-2-3, 1-2-4, 1-3-4, 2-3-4).

For Figures 9(a) and 9(b), we first estimated speed and length for each signature using the mean of all four estimates. As can be seen, use of multiple estimates greatly reduces measurement variance. Variance is larger when speeds are faster at the BN site. Finally, variation in length estimates is considerably less than variation in speed. Three-category classifications are affected by even small variations, however, since there are many vehicles near the dividing line between P and S*.

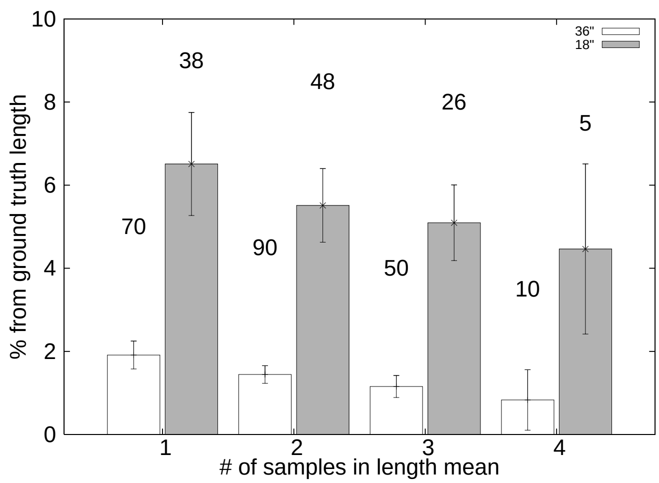
To test against ground truth, we took an additional, custom experiment when we ran a vehicle with known length across sensors spaced at 18 and 36 inches (details of the experiment are in Section 6.1).



(a) Speed compared to four estimates.



(b) Length compared to four estimates.



(c) Length compared to ground truth.

Figure 9: Comparison of the use of multiple estimates from a single signature. The first two use data from sites A (white bars) and BN (gray bars) of the USC experiment, while the final figure uses data from the first ISI rooftop experiment.

Table 8: Classification Results, Trucks vs Non-Trucks

Site		Total considered	Correctly classified	Unable to classify [segmentation errors]	Incorrectly classified
A	Total	164 (100%)	133 (81%)	16 (10%) [9 SE]	15 (9%)
	T	8	6	2	0
	Non-T	156	127	14	15
BN	Total	248 (100%)	212 (85%)	18 (7%) [13 SE]	18 (7%)
	T	37	30	3	4
	Non-T	211	182	15	14

Table 9: Three-category Classification Results

Site		Total considered	Correctly classified	Unable to classify [segmentation errors]	Incorrectly classified			
					Total	Wrong as P	Wrong as S*	Wrong as T
A	Total	164 (100%)	94 (57%)	16 (10%)[9 SE]	54 (33%)			
	P	101	72	9	20	–	9	11
	S*	55	16	5	34	30	–	4
	T	8	6	2	0	0	0	–
BN	Total	248 (100%)	174 (70%)	18 (7%)[13 SE]	56 (23%)			
	P	127	90	13	24	–	17	7
	S*	84	54	2	28	21	–	7
	T	37	30	3	4	0	4	–

Figure 9(c) shows these results. The trend in the right graph against known ground truth matches the trend in the middle graph: more estimates reduce variability. However, this experiment also confirms that less variability corresponds with more accuracy.

6.2.2 Sensor-Based Classification

We next consider sensor-based classification using data collected from both A and BN sites.

As we discussed in the SURE-SE final report, we manually deleted records of motorcycles, carts, and bicycles from the dataset. It should not be hard to eliminate these automatically, but our current focus is on the harder problem of distinguishing cars from trucks, not the easier problem of cars from bicycles. As future work we plan to automate this filtering. After filtering we were left with 164 records on site A and 248 records on site BN. In addition, we found 22 records (9 on site A and 13 on site BN) were missing from sensor data set, due to segmentation errors. Although we do not yet automatically detect these, we believe we can do so relatively easily. We therefore report these values as "segmentation error (SE)" on Table 8 and 9.

We started by classifying vehicles in two groups: trucks (including 18-wheelers, cement mixers, and panel-trucks) and non-trucks (includes everything else other than trucks). Results are shown in Table 8. With just two categories (trucks, type T, vs. non-trucks), our classification rates are comparable to current, state-of-the-art published results from Sun *et al.* [24]. These results were particularly encouraging given that our system was not tuned to specifically deal with the high variability in vehicle speed that we found.

In addition to the two categories of vehicles mentioned above, we expanded our initial classification algorithm to include a third category, S*, comprised of SUVs, pick-up trucks, vans and minivans. Results are shown in Table 9. Our accuracy with three categories is not as high as with only two categories. Our algorithms depend primarily on vehicle length, but the vehicles in class S* and P overlap in length. No classification algorithm based only on length will be able to separate these categories accurately. This can be seen in the data in Table 9 where many "medium-size" vehicles are incorrectly classified as passenger

Table 10: Effect of careful feature extraction (single sensor)

Site	**simple** (peaks)	**careful feature** (edges) without consistency	**careful feature** (edges) with consistency
Two-category classification			
A	110 (67%)	133 (81%)	133 (81%)
BN	202 (81%)	209 (84%)	212 (85%)
Three-category classification			
A	55 (34%)	93 (57%)	94 (57%)
BN	157 (63%)	164 (66%)	174 (70%)

cars (51 out of 139). By comparison, classification of trucks is quite good, with very few trucks being misclassified as S*, and very few S* or P types being classified as T.

This experiment suggests, first, that our approach is very appropriate if the goal is to classify trucks from non-trucks. For studies about road damage, this level of classification may be sufficient. It also suggests that very fine-grain classification of passenger cars, vans, pickups, SUVs will be quite difficult, given the blending of these vehicle types.

It is useful to compare the accuracy of our sensor network-based classification system to manual (human) classification described previously. First, considering just counting, manual classification was 83–87% accurate. By comparison, our classification system had accuracy rates of 81–85% for two-category classification and 57–70% for three-category classification. From this we conclude that our single-sensor system is comparable to humans for two-category counts, because our system can handle vehicles as rapidly as they occur, while humans can become overloaded and make errors when too many vehicles appear quickly.

We also must state that for three-category classification, humans remain more accurate when compared to our current system. This result is because, even though manual counts miss many vehicles, humans are much better at distinguishing “SUV-like” vehicles (type S*) than our system which uses simple length-based measures.

Finally, it is important to note that these experiments evaluate classification between the most challenging FHWA classes (cars from trucks). Future work might include evaluation of our approach over simpler FHWA classes, such as different kinds of semi-trailer configurations.

6.2.3 Effect of Careful Feature Extraction

We earlier discussed the importance of using distinct features, and of detecting common problems such as channelization error (Section 4.2.2).

To quantify the benefits of our improved feature detection Table 10 compares overall classification accuracy of two and three-category analysis with and without these improvements. The three columns compare detection of simple peaks only, with detection of careful features (edges), with and without consistency checking.

As can be seen, the shift from peaks to edges is helpful at both sites. It is particularly helpful at site A (improving accuracy 14% for two-category and 23% for three-category) because there speeds are generally lower (8mph compared to 16mph at site BN), making wheel peaks much less distinct because of longer time spent over the sensor.

Consistency checking is most helpful at site BN (improving accuracy 1% and 4%) because that sensor suffered from a number of missing wheel cases. Site A suffers from a number of channelization errors which we detect but cannot correct. This detection does alter the quality estimate, however, allowing multi-sensor techniques to select the estimate without channelization error as described below.

Table 11: Multi-sensor classification accuracy.

Total vehicles at both A and BN:	39 (100%)
Two-category classification	
single sensor:	
A alone:	36 (92%)
BN alone:	36 (92%)
multi-sensor combining A and BN:	
oracle:	38 (97%)
shortest-best:	38 (97%)
quality-best:	38 (97%)
Three-category classification	
single sensor:	
A alone:	24 (61%)
BN alone:	27 (69%)
multi-sensor combining A and BN:	
oracle:	32 (82%)
shortest-best:	25 (64%)
quality-best:	29 (74%)

6.2.4 Use of Sensor Fusion

Finally, we wish to investigate our hypothesis that multiple sensors can help resolve independent errors. In our experiment, 39 vehicles passed through both sites A and BN. We compare our two sensor fusion algorithms from Section 4.2.3, shortest-best and quality-best, with an oracle algorithm. Recall that the oracle algorithm takes the correct classification if either individual sensor is correct, thus providing a theoretical upper bound on performance.

Table 11 summarizes the results of this comparison, showing that our sensor fusion algorithms can correct several kinds of independent errors such as mis-segmentation and mis-channelization (described in Table 5). For two-category classification, both shortest- and quality-best always select the correct classification, matching the oracle. We cannot achieve more than the oracle unless we enhance individual sensor accuracy.

In three-category classification, the quality-best algorithm improves the accuracy by 5–14%. This improvement suggests that the confidence value we used for quality-best algorithm captures accuracy of the individual sensor's estimation and helps us to select the better classification. The shortest-best algorithm does not do as well as quality-best because it is designed to correct only segmentation errors and does not attempt to consider other kinds of errors.

Reviewing the errors from Table 5, we addressed mis-channelization error by detecting it at single sensors (Section 6.2.3), then selecting the best quality single-sensor with quality-best (Section 6.2.4). We handled imprecise speeds by selecting an appropriate sensor spacing and sampling frequency (Section 6.1). Changing speeds are best handled in single sensors. Segmentation errors are addressed by the shortest-best and quality-best sensor fusion algorithms (Section 6.2.4).

6.3 Verification of Autoconfiguration Algorithms

We wanted to verify the effectiveness of our autoconfiguration algorithms described in Section 4.3. To do this we set up a small experiment on an ISI rooftop (Section 5.3). In that experiment, the initial threshold works perfectly after a single pass of auto-configuration.

We plan to exercise this algorithm further in future experiments to further validate this algorithm.

7 Plans and Future Work

This project has demonstrated that automated systems can classify vehicles as accurately as humans provided a sensor network can combine readings from multiple sensors. In principle, such a system could be assembled from parts costing only a few thousand dollars, and deployed with only a few minutes of traffic interruption.

However, several open challenges remain before such a system could be given to students, researchers, or traffic engineers. First, we must understand how to connect a cluster of in-the-field traffic sensors to each other and to a researcher's central database or traffic management system with moderate cost. Second, we must integrate and automate deployment of the system as a whole. Finally, significant work is required to move from demonstrating the principle behind a sensornet for traffic monitoring to having an integrated system that can be taken into the field.

We must to develop approaches to economically connect traffic sensors to each other and to central researcher or traffic management systems. The key constraints on in this potential work are to do so *economically* with *traffic sensors* in-the-field.

Custom industrial wireless networks are widely used today for many applications in specific locations, but the need to lease radio spectrum and develop or purchase custom radios make custom approaches very costly. Cellular telephone and data networks also provide wide coverage, but the cost is quite high, often $80 per computer per month, making their use uneconomical for dozens of sensors in a sensor network. In addition, they are quite energy hungry, running only a few days on idle, and a few hours of use.

We also must understand the unique demands of a network of traffic sensors. The requirement that a traffic sensornet be easily deployed means we wish to allow computers to operate on battery power, suggesting the use of low-power radios and network protocols, such as 802.15.4 and S-MAC. These radios provide throughput of only 50kb/s or less, comparable to telephone modems.

Our second goal is to automate deployment of the system as a whole. Our work in SURE-FT helped define the parameters that need to be automatically configured in our traffic classification network: basic sensor calibration, loop width, spacing between pairs of loops, relative locations of loops that will see the same traffic, and several parameters relating to signature interpretation (the number of samples allowed in a wheel-well "dip" and the thresholds for wheel start and steep-slope start). This work also indicated that classification results are relatively insensitive to starting thresholds, provided we use algorithms that detect and handle partial signatures. The physical loop layout parameters are still important, so calibrating these is essential.

Finally, significant work is required to move from demonstrating the principle behind a sensornet for traffic monitoring to developing an integrated system that can be taken into the field.

These areas are potential directions for future research.

8 Conclusions

This report summarizes the research tasks of the SURE-FT project. The main conclusions of this work were to verify that one can effectively use multiple sensors to reduce error rates in vehicle classification, and to begin the process of developing an integrated system that can be easily deployed and automatically configured to collect such data.

9 Acknowledgments

We are grateful for the assistance of Mr. Steven Hilliard, IST Corporation, the Los Angeles City Department of Transportation, and University of Southern California, Transportation Services Division. Several graduate students participated in the field data collection.

We thank Padma Haldar, Arnold Diaz, and Dongjin Son for use of their person automobiles for additional tests, and Dongjin Son for assisting in ISI rooftop experiments.

References

[1] National Highway Traffic Safety Administration. NHTSA Wheelbase and track width information. `http://www.nhtsa.dot.gov/`.

[2] D. Banister and J. Berechman. *Transportation Investment and Economic Development*. UCL Press, London, UK, 2000.

[3] Richard R. Brooks, Parmesh Ramanathan, and A. M. Sayeed. Distributed target tracking and classification in sensor networks. *Proceedings of the IEEE,* 91(8):1163–1171, August 2003. Invited Paper.

[4] Sing Yiu Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. In *Proceedings of the Annual Meeting of the Transportation Research Board*. Transportation Research Board, January 2005.

[5] Sing Yiu Cheung, Sinem C. Ergen, and Pravin Varaiya. Traffic surveillance with wireless magnetic sensors. In *Proceedings of the 12th World Congress on Intellgient Transport Systems*, San Francisco, California, USA, November 2005.

[6] Intel Stargate Computer. Crossbow Technologies SPB400.

[7] Prabal Dutta, Mike Grimmer, Anish Arora, Steve Bibyk, and David Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proceedings of the Fourth IEEE International Workshop on Information Processing in Sensor Networks*, pages 497–502, Los Angeles, California, USA, April 2005. IEEE.

[8] Federal Highway Administration and Office of Highway Policy Information. FHWA vehicle types. web page `http://www.fhwa.dot.gov/policy/ohpi/vehclass.htm` and `http://www.dot.state.oh.us/techservsite/availpro/Traffic_Survey/SchemeF%/FHWA_Scheme_F_Report.PDF`, October 2003.

[9] Ford. Ford Vehicle. `http://www.fordvehicles.com/`.

[10] Genevieve Giuliano, Peter Gordon, Qisheng Pan, Jiyoung Park, and Lanlan Wang. Estimating freight flows for metropolitan area highway networks using secondary data sources. submitted for review to Networks and Spatial Economics, 2006.

[11] Genevieve Giuliano, T. O'Brien, and J. Maggadino. Evaluation of terminal gate appointment system at los angeles and long beach ports. Technical Report Final Report Project 04-06, USC/CSLB METRANS Transportation Center, 2005.

[12] Peter Gordon and Qisheng Pan. Assembling and processing freight shipment data: Developing a gis-based origin- destination matrix for southern california freight flows. Final report, National Center for Metropolitan Transportation Research (METRANS), 2001.

[13] Lin Gu, Dong Jia, Pascal Vicaire, Ting Yan, Liqian Luo, Ajay Tirumala, Qing Cao, Tian He, John A. Stankovic, Tarek Abdelzaher, and Bruce H. Krogh. Lightweight detection and classification for wireless sensornetworks in realistic environments. In *Proceedings of the Third ACM SenSys Conference*, pages 205–217, San Diego, California, USA, November 2005. ACM.

[14] M. Hansen, U A. Dobbins D. Gillen, Huang, and Puvathingal M. The air quality impacts of urban highway expansion: Traffic generation and land use change. Technical Report UCB-ITS-RR-93-5, Institute of Transportation Studies, University of California, Berkeley, CA, USA, 1993.

[15] John Heidemann, Fabio Silva, Xi Wang, Genevieve Giuliano, and Mengzhao Hu. *SURE-SE: Sensors for Unplanned Roadway Events—Simulation and Evaluation: Final Report*. USC/Information Sciences Institute, May 2005.

[16] Inductive Signature Technologies Inc. IST sensor model 222.

[17] Lawrence Klein. Final report: Mobile surveillance and wireless communication systems field operational test; volume 1: Executive summary. Technical Report UCB-ITS-PRR-99-6, University of California, Institute of Transportation Studies, March 1999.

[18] Ara N. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. Master's thesis, Massachusetts Institute of Technology, June 2000.

[19] M.D. Meyer and E.J. Miller. *Urban Transportation Planning: A Decision-Oriented Approach*. McGraw-Hill, second edition edition, 2001.

[20] E. Miller, D. Kriger, and J. Hunt. Research and development program for integrated urban models. In *Proceedings of the Annual Meeting of the Transportation Research Board*, pages 169–176. Transportation Research Board, January 1999.

[21] Cheol Oh, Stephen G. Ritchie, and Shin-Ting Jeng. Vehicle reidentification using heterogeneous detection systems. In *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington, DC, USA, January 2004. Transportation Research Board.

[22] E. Pas. *The Geography of Urban Transportation, second edition, S. Hanson, editor*, chapter The urban transportation planning process. The Guilford Press, 1995.

[23] A. Sivakumar and C. Bhat. A fractional split distribution model for statewide commodity flow analysis. Technical report, University of Texas, Austin Department of Civil Engineering, Austin, TX, USA, 2002.

[24] Carlos Sun, Stephen G. Ritchie, and Seri Oh. Inductive classifying artificial network for vehicle type categorization. *Computer-Aided Civil and Infrastructure Engineering*, 18(3):161–172, 2003.

[25] Transportation Research Board. Expanding metropolitan highways. Special Report 245, National Research Council, Washington, DC, USA, 1995.

[26] Transportation Research Board. Evaluation of the congestion mitigation and air quality program. Special Report 256, National Research Council, Washington, DC, USA, 2002.

[27] Xi Wang, Fabio Silva, and John Heidemann. Infrastructureless location aware configuration for sensor networks. Technical Report ISI-TR-2004-589, USC/Information Sciences Institute, June 2004.

[28] M. Wegener. Operational urban models: State of the art. *Journal of the American Planning Association*, 60(1):17–29, 1994.

[29] Feng Zhao, Jaewon Shin, and James Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.