ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221234378

The Pothole Patrol: Using a mobile sensor network for road surface monitoring

Conference Paper · January 2008

DOI: 10.1145/1378600.1378605 · Source: DBLP

CITATIONS
973

READS
2,607

6 authors, including:

Jakob Eriksson
University of Illinois at Chicago
35 PUBLICATIONS 4,008 CITATIONS

SEE PROFILE

Lewis Girod
Massachusetts Institute of Technology
86 PUBLICATIONS 9,820 CITATIONS

SEE PROFILE

Ryan Newton
Massachusetts Institute of Technology
154 PUBLICATIONS 2,409 CITATIONS

SEE PROFILE

Samuel Madden
Massachusetts Institute of Technology
378 PUBLICATIONS 37,137 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Spinal codes View project

Status Quo View project

All content following this page was uploaded by Jakob Eriksson on 29 January 2014.

The user has requested enhancement of the downloaded file.

The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring

Jakob Eriksson, Lewis Girod, Bret Hull,
Ryan Newton, Samuel Madden, Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory
{jakob,girod,bwhull,newton,madden,hari}@csail.mit.edu

ABSTRACT

This paper investigates an application of mobile sensing: detecting and reporting the surface conditions of roads. We describe a system and associated algorithms to monitor this important civil infrastructure using a collection of sensor-equipped vehicles. This system, which we call the *Pothole Patrol* (P^2), uses the inherent mobility of the participating vehicles, opportunistically gathering data from vibration and GPS sensors, and processing the data to assess road surface conditions. We have deployed P^2 on 7 taxis running in the Boston area. Using a simple machine-learning approach, we show that we are able to identify potholes and other severe road surface anomalies from accelerometer data. Via careful selection of training data and signal features, we have been able to build a detector that misidentifies good road segments as having potholes less than 0.2% of the time. We evaluate our system on data from thousands of kilometers of taxi drives, and show that it can successfully detect a number of real potholes in and around the Boston area. After clustering to further reduce spurious detections, manual inspection of reported potholes shows that over 90% contain road anomalies in need of repair.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]:

General Terms

Algorithms, Design, Experimentation, Measurement

1. INTRODUCTION

Municipalities around the world spend millions of dollars to maintain and repair their roadways.[1] Despite this investment, few people are happy with the quality of the roads where they live or work. The reason is that bad roads damage vehicles, are sometimes hazardous to drivers and pedestrians, and, at the very least, are annoying to drive or bike on. They are also the cause of expensive lawsuits and damage claims—for example, in 2005, the state of Michigan had more than 7,500 pothole-related damage claims filed against it,[2] and insurance companies receive more than 500,000 pothole-related claims each year.[3]

Keeping our roadways in good condition is a challenging problem because harsh weather, unexpected traffic load, and normal wear and tear all degrade even well-laid roads over relatively short periods of time (weeks to months). Because municipal budgets are generally constrained, determining *which* roads need fixing becomes important. In addition, informing drivers of hazardous road conditions especially at night or when lighting is poor would be a useful feature for navigation systems. This paper seeks to address this need.

We describe the design, implementation, and experimental evaluation of Pothole Patrol (P^2), a mobile sensor computing system to monitor and assess road surface conditions. P^2 uses three-axis acceleration sensors and GPS devices deployed on embedded computers in cars, relying on the inherent mobility of cars to traverse the roads being monitored. This opportunistic mobility is well-suited to the application at hand for three reasons. First, it is cost-effective when deployed on taxis, garbage trucks, postal vehicles, volunteers' cars, etc. Second, it achieves high spatial coverage with even a small number of cars (in our data, 7 cabs are able to cover 2492 distinct kilometers during their normal driving in 10 days). Third, it is more systematic and reliable than previous approaches to the problem (§6) because it assesses the conditions of any road segment using multiple drives from multiple collaborating cars.

Road surface monitoring is a problem that fundamentally requires *mobility* to solve; it cannot easily be solved by deploying static sensors on the roads. In addition to the sheer size of the roadway network that would make a static sensor deployment daunting in terms of labor and cost, road conditions are naturally sensed from a moving entity that can measure vibrations and impulses during a drive. Luckily, because roads deteriorate on time-scales of many weeks, they need to be sampled at most a few times per day (or even less). With the addition of inexpensive, easy-to-install hardware, gathering sensor data for road surface monitoring becomes a useful side-effect of normal vehicular operation.

At a high level, the operation of the P^2 system is simple. Each vehicle's embedded device gathers three-axis acceleration data at a high frequency (380 times per second in our implementation) and position data once per second. On-board signal processing software takes the vertical z-axis and sideways x-axis acceleration samples and the reported speed, yielding a low-rate stream of of high-probability pothole detections. The embedded node in the ve-

[1]http://boston.bizjournals.com/boston/othercities/denver/stories/2007/04/02/story1.html?b=1175486400%5E1438887

[2]http://www.detnews.com/2005/specialreport/0510/18/A01-350197.htm

[3]http://www.wktv.com/special/6733696.html

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiSys'08, June 17–20, 2008, Breckenridge, Colorado, USA.
Copyright 2008 ACM 978-1-60558-139-2/08/06 ...$5.00.

hicle delivers these detections to a central server over a wireless network, using our reliable delay-tolerant pipe mechanism, $dPipe$. The server maintains a database of detections from multiple vehicles, and combines these to produce a set of "road anomalies", including potholes and other rough road conditions.

This simple description hides three important problems that P² must solve. First, there are a large number of benign events, including braking, doors being slammed, sudden swerves, etc. that all yield high-energy acceleration signatures. In addition, there are other road anomalies like expansion joints and railway tracks that are even harder to distinguish from genuine potholes. The second problem is that it can be difficult even for a human to tell whether any given road anomaly is really a pothole that merits fixing, or just a "bump" in the road . As a result, ground truth is hard to obtain and requires a careful experimental method. The third problem is that a given pothole or anomaly does not manifest the same way during each drive over it; the values reported by the sensors depend on how the car approached the pothole, its speed, and how the sensors are mounted on the car. Coping with these variations requires a consistent approach to mounting the sensors and developing a robust set of features from the sensor observations.

To address these problems, we use a machine learning approach. We begin by manually collecting a set of training data by carefully and repeatedly re-driving a set of roads and noting the location of various different classes of road anomalies (potholes, railroad crossings, etc.) A set of training samples is extracted based on the manually determined location of such road anomalies, and each sample is labeled with the type of anomaly that occurred in it.

We then train a detector based on peak X and Z acceleration, as well as the instantaneous velocity of the vehicle, to maximize correct detections, while minimizing false positives. To improve the robustness of the detector, we use a set of "loosely labeled" data which is a low-effort means of extending the set of training data to a broader set of roads. Finally, we added a clustering-based filter, which, when run on the entire set of data from our taxis data set, only reports those detections which occurred more than a few times; this method allows us to filter out spurious events, which may not be due to a road anomaly.

The contributions of this work are:

- Our system, P², is the first (to our knowledge) mobile end-to-end system for detecting and reporting potholes and other road anomalies. It currently runs on a network of 7 taxis in the Boston metropolitan area.

- We develop a pothole detection system that is able to successfully differentiate potholes from a variety of other road anomalies using accelerometer data recorded during normal driving by a set of taxis. Our detector has a very low false-positive rate for potholes, flagging less than 0.2% of samples from good roads as potholes on our test data set.

- We show experimentally that our system, when run on 9730 (of them, 2492 unique) kilometers of real-world taxi traces, is able to successfully identify dozens of potholes and other anomalies. After clustering to reduce spurious detections, we verified through manual inspection that over 90% of potholes reported by our P² deployment contained potholes or other road anomalies in need of repair.

2. P² ARCHITECTURE

In this section, we describe the software architecture of the P² system as well as the sensors used to capture data and the and the testbed used to evaluate it.

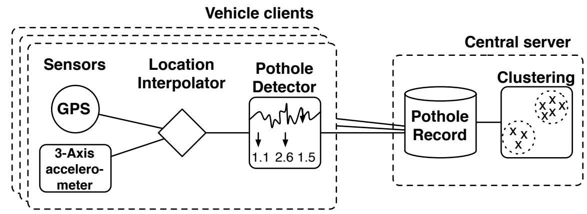

Figure 1: P² road monitoring architecture. Cars process the raw data to produce detections, which are reported to a central server, where clustering filters out spurious detections.

P^2 consists of a set of sensor-equipped vehicles, and a central server, as illustrated in Figure 1. Raw sensor data is collected using a GPS device (1 Hz) and a 380-Hz accelerometer, resulting in the following information:

```
<time,location,speed,heading,3-axis
acceleration>
```

The first four parameters come from the GPS device and the acceleration vector comes from the three-axis accelerometer. These two data streams are combined using GPS interpolation. On-board processing filters the combined data stream to produce high-probability pothole detections. When network connectivity is available, the cars automatically upload their detections to a central server, which maintains a database of detections. The central server clusters detections based on location, and applies a minimum cluster size, resulting in the final output of the system: a series of "pothole" detections of varying confidence and severity.

2.1 Connectivity and Delay Tolerance

We upload detections to our central servers using opportunistic WiFi connections provided by participating open WiFi access points, or using a cellular data service, where available. Although cellular data service offers more reliable connectivity than WiFi, even this mode of connectivity frequently experiences pockets of no coverage. To cope with such outages, we implemented a local buffering and reliable transmission scheme, called *dPipe*. dPipe is a conceptual extension to the UNIX pipe abstraction that allows processes on separate hosts to communicate via a reliable, delay-tolerant data stream. It is implemented using several file-based buffers for storage, and, when connectivity is present, uses TCP sockets to send buffered data and application-level acknowledgments to ensure that all data gets written to disk.

2.2 Taxi Testbed

For our P² deployment, we used a mobile testbed that runs on local taxis. In exchange for some fleet management facilities, the taxi company has allowed access to their cars for our experiments. Each taxi in the testbed (which currently contains 7 cars) is equipped with a Soekris 4801 embedded computer running Linux, a WiFi card, a Sprint EVDO Rev A network card, an external GPS (mounted on the roof of the car), and a 3-axis accelerometer (mounted in the same place and with the same orientation in each car). In our experiments, all the cars are Toyota Priuses, from several different model years; different drivers drive different cars on different days.

In our experiments, these taxis traveled over 2492 distinct kilometers of roads in the greater Boston area and 9730 kilometers in total. Even with only 7 cabs deployed over a period a few weeks, 174 km of road was covered with ten or more repeated passes; such repetition improves the reliability of detecting potholes and other

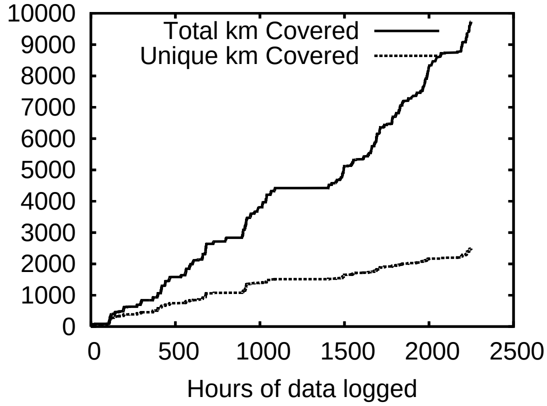

Figure 2: Distance traveled vs. total hours driven across all taxis. The lower line represents total *unique* roads encountered (discounting repeated observations of the same road).

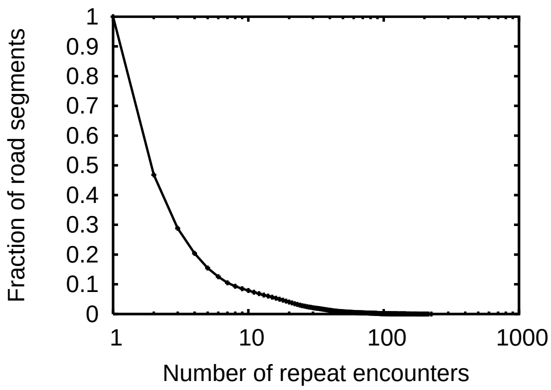

Figure 3: CDF of repeat-coverage of road segments. Includes $258,021$ unique road-segments each approximately 10 meters in length.

anomalies. Figure 2 shows the total kilometers of driving against hours of data collected. Note that for many of these hours, the cabs are standing still. Also note that the estimated length of unique road traversed does not count lanes going in different directions on the same road separately. Figure 3 shows the distribution of coverage density across all the lengths of road encountered.

3. DATA ACQUISITION

Our system depends on the accelerometer producing consistent results for a given pothole, and on having accurate localization of events from the on-board GPS. In this section, we describe a few experiments we performed to validate the functioning of our sensors. We also discuss how our training data was gathered.

3.1 Accelerometer Placement

One concern is that the placement of accelerometers inside the vehicle might affect the quality of the signal. To answer this question, we placed accelerometers in three locations inside the cabin of a single car. Figure 4 shows the accelerometer signal for a fixed stretch of pavement from three different mounting positions: attached to the dashboard, attached to the right side of the windshield, and attached to the embedded PC, which was not firmly attached to the vehicle. We tested the last position because it is easy to install. The signals from the dashboard and windshield appear to be quite similar, while the accelerometer attached to the computer produced unpredictable results. Consequently, we firmly attached the accelerometer to the dashboard inside the car's glove box, which is a relatively easy location to install sensors on, and which keeps the sensors out of the way of passengers in the cabin.

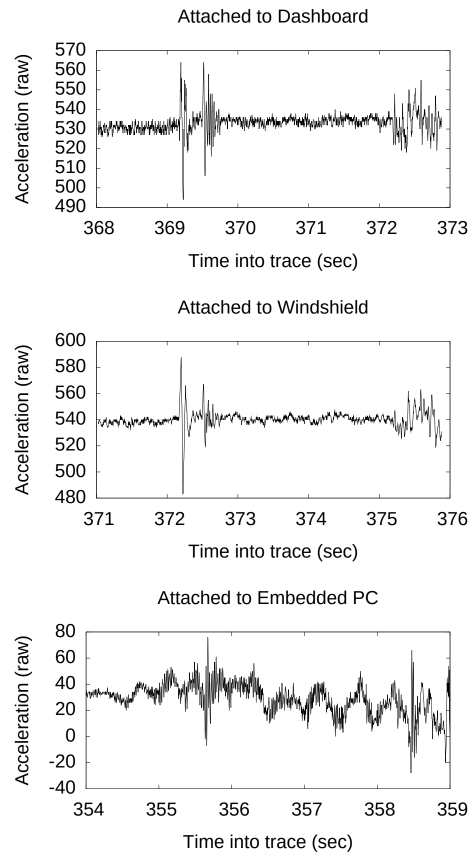

Figure 4: These graphs show the how the accelerometer signal (z axis) varies with placement inside the car. Each graph shows a different placement: either attached to the windshield, dashboard, or attached to the embedded PC. We found firm attachment to the dashboard generated the cleanest signal while still being easy to install.

3.2 GPS Accuracy

GPS accuracy in our deployment is important if potholes are to be properly located and multiple detections combined to report a single pothole. To measure accuracy, we placed a thick metal bar across a road, and repeatedly drove over it. For each drive, we first identify the peak accelerometer reading r in the drive, and then find the estimated location of the car when r occurred, using linear interpolation between GPS readings. We found the standard deviation of the positions reported for the bar to be 3.3 meters, which is consistent with typical measurement errors from modern GPS receivers outdoors.

3.3 Hand-labeled Training Data

We collected the hand-labeled data by repeatedly driving down several known stretches of road in the Boston area, and continuously recording raw accelerometer traces. A passenger in the car

Type	Count	Percentage
Smooth road (SM)	64	23%
Potholes (PH)	63	23%
Manholes (MH)	59	21%
Railroad Crossing (RC)	18	6%
Crosswalk/Exp. Joint (CWEJ)	76	27%

Table 1: Distribution of hand-labeled training data.

labeled each event encountered in real-time by pressing a key on a laptop every time the impact of one class of event was felt. Traces were post-processed to select out only the sample windows containing a corresponding event that appeared significant, in order to eliminate delay and inaccuracy in the human-recorded annotations.

We focused on collecting a diverse set of samples, including the following event classes:

- **Smooth road (SM)**: Segments of road surface that are considered smooth.
- **Crosswalks and Expansion Joints (CWEJ)**: Crosswalks using extra-thick paint, brick, strips of pavers, or raised dots. Metal expansion joints in bridges and overpasses.
- **Railroad Crossing (RC)**: Train tracks. Such crossings can be jarring, and are sometimes confused for a disturbed road surface.
- **Potholes (PH)**: Missing chunks of pavement, severely sunk in or protruding manhole covers, other significant road surface anomalies.
- **Manholes (MH)**: Manhole covers and other equipment in the road that are nearly flush with the road surface. Moderate cracking, sinking or bulging.
- **Hard Stop (ST)**: A rapid deceleration, sometimes with the familiar jerk at the end.
- **Turn (TU)**: Turning a corner. This sometimes exhibits a rather violent acceleration profile.

We sometimes refer to the set of classes excluding smooth road as "road anomalies".

For the smooth road samples we select a number of windows from segments of data that we have manually determined to be free of anomalies. Table 1 shows the number and distribution of different labels for our hand-labeled set. This data set is not intended to be representative of the true frequencies of various classes of roads in practice (where, for example, smooth road is much more prevalent), but instead designed to create a diverse set of training data.

3.4 Loosely Labeled Training Data

Relying only on carefully hand-labeled datasets severely limits the amount of data available for training. In particular, our hand-labeled coverage does not have a broad enough coverage of the non-pothole types of road surfaces, resulting in a tendency to over-report pothole detections when faced with new data. To address this issue we collected several traces of "loosely labeled" data. A loosely labeled dataset is one in which the types and rough frequency of anomalies present in the data is known, but their exact number and location in the trace is not. For example, a given trace might be known to contain some distribution of manholes, expansion joints, and smooth road, but no potholes.

- **Storrow Dr.** Heavily used four-lane parkway on the Boston side of the Charles River with several bridges, some potholes.
- **Memorial Dr.** Heavily used four-lane parkway on the Cambridge side of the Charles River, good condition.
- **Binney St.** A two-lane street with many sunk-in manholes and sealed cracks, one pothole.
- **Hwy I-93** An 8 lane interstate highway that cuts through the center of Boston in good condition.
- **Beacham St** A heavily trafficked back road in very poor condition.

Table 2: Loosely labeled road segments. For these roads, we estimate number and types of event classes, but do not record their exact location and type.

Our loosely labeled data was collected from the road segments described in Table 2. Loosely labeled data, when used judiciously, can be used to extend the training set and further improve detector performance, as we will see in §5.

4. POTHOLE DETECTION ALGORITHM

In this section, we describe the algorithm we have developed to detect road anomalies in streams of sensor data.

The intuition behind our algorithm is that anomalous road conditions are reflected in features of the acceleration data. The problem of identifying potholes from accelerometer data is challenging because of the broad variation in road conditions (e.g., various types of road surfaces and anomalies such as potholes, manholes, curbs, railroad crossings, and expansion joints) and driver behavior (turning, swerving, sudden braking, etc.) While most anomalies can be characterized as high-energy events in the acceleration signal, signal energy content alone is not sufficient as a detection criterion, because many high-energy events should not be considered road anomalies. For example, as illustrated by the sample accelerometer traces in Figure 5, road fixtures such as railroad crossings and expansion joints can generate significant acceleration impulses, and high energy events can be caused by passengers when they slam the door or drivers braking suddenly.

In P^2, acceleration data is processed as it is received by the embedded computer in the vehicle. The trace is segmented into 256-sample windows; the events we are interested in are generally of even shorter duration. Figure 6 illustrates how these windows are processed to detect the presence of a pothole impact. A series of signal processing filters are then applied to this continuous stream of windows, where each filter is designed to reject one or more non-pothole event types. We now discuss each filtering stage in more detail.

1. **Speed**: Windows in which the car is not moving, or is moving very slowly, are ignored. This stage rejects events such as door slams, as well most curb ramps.
2. **High-pass**: The high-pass filter removes low-frequency components from the acceleration signal in all the x- and z-axes. Such components can be introduced by acceleration, turning, veering, braking, and as well as subtle changes in sensor orientation.
3. **z-peak**: Peak acceleration in the z-axis is a prime characteristic of significant road anomalies, as seen in Figure

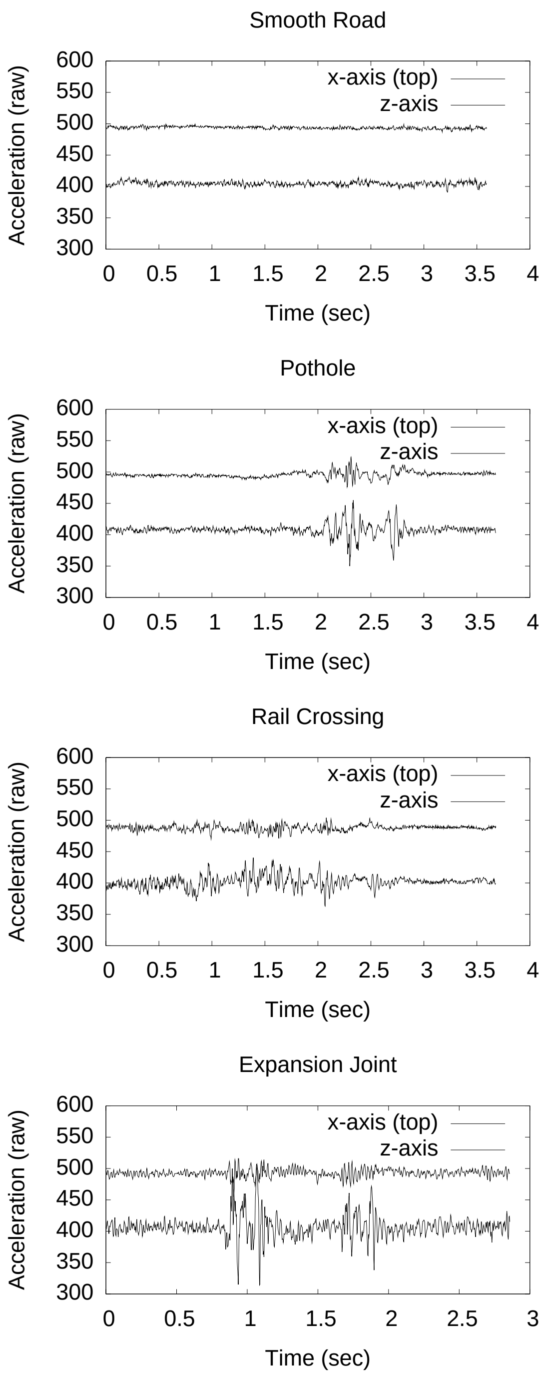

Figure 5: Sample accelerometer traces of common event types.

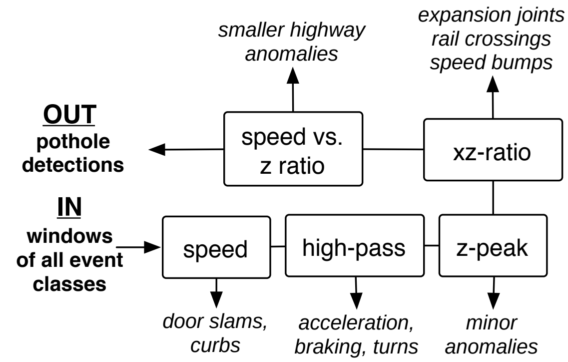

Figure 6: The pothole detector is composed of a number of filters, each separating out a different class of event.

5. This filter rejects all windows with a peak (absolute) z-acceleration less than a threshold t_z.

4. **xz-ratio**: Acceleration in the x-axis can help identify road anomalies that span the width of the road (such as railway crossings, speed bumps and expansion joints), and therefore impact both sides of the car equally. Assuming potholes only impact one side of the car, a true pothole event with a large z-peak acceleration should produce a significant peak in the x-axis within some small time window Δw. Compare the "pothole" trace to the "expansion joint" trace in Figure 5. Here the expansion joint produces a much larger peak z-acceleration than the pothole, yet this is accompanied by a relatively smaller x-acceleration. This filter rejects windows where the peak x-acceleration within Δw samples (we use $\Delta w = 32$) from the peak z-acceleration reading, is less than some factor t_x times the peak z-acceleration.

5. **speed vs. z ratio**: At high speeds, even small road anomalies can create high peak acceleration reading. This filter rejects windows where the peak z acceleration is less than a factor t_s times the speed of travel.

4.1 Training the Detector

The last three filters are controlled through tuning parameters $\mathbf{t} = \{t_z, t_x, t_s\}$, which control the thresholds for rejecting an event as a "non-pothole". The values of these parameters are computed using a combined exhaustive search over a reasonable range of values for each parameter. For each set of parameter values $\mathbf{t}$, we compute the detector score $s(\mathbf{t}) = corr - incorr^2$. Here, $corr$ is the number of pothole detections reported when using $\mathbf{t}$ where the training samples were in fact labeled as "pothole", and $incorr$ is the number of reported pothole detections where the training sample was labeled as belonging to some other class. We use the square of the number of incorrect detections to emphasize the importance of minimizing false positives. The final parameter set is chosen to maximize $s(\mathbf{t})$. Since this computation can be done offline, and is only done based on a small set of training data, computational complexity is not a concern here. However, an iterative refinement technique could be used to speed up this process, if required.

To improve the robustness of the detector, the training phase also accepts traces from the loosely labeled data, which consists of entire roads marked roughly with many potholes exist in each stretch of road. By not requiring training data to be carefully extracted, labeled and cleaned, we can dramatically increase the amount of training data available to the classifier. In contrast with the hand-labeled training data, loosely labeled data does not reveal exactly where and when a pothole was encountered. This makes it difficult to determine whether a detection is correct or not. On the other hand, loosely labeled data does provide a clue as to how many detections, $count_r$ to expect for a given stretch of road, r. The following detector score function makes use of this quantitative information available in loosely labeled training data

$$s(\mathbf{t}) = corr - incorr_{labeled}^2 - \max(0, incorr_{loose} - count_r).$$

The $\max$ term ensures that no negative score is assigned until the number of detections exceeds the expected number. While loosely labeled training data is more plentiful than hand-labeled data, it is also less reliable, hence the square term for errors on hand-labeled training data, and the linear term for loosely labeled training data.

4.2 Clustering By Location

The pothole events reported by the detector are likely to include some false positives. To improve accuracy, we require possible events to be corroborated to be considered valid, meaning that a cluster of at least k events happen in the same location (with a small margin of error Δd), while moving in the same direction. This spatial corroboration helps filter out misclassified events that are unrelated to spatial location (and hence are definitely not potholes), such as vibrations from within the vehicle—as it is unlikely that such events would happen multiple times in the exact same location.

The following algorithm is used to compute clusters of pothole detections. Starting with the location of each candidate road surface anomaly detection, we first compute pairwise distances, for all distances $< \Delta d$. This can be done efficiently by first placing the detections in a $\Delta d \times \Delta d$ grid, and computing pairwise distances only between detections in the same or neighboring grid cells. Clusters are then formed by iteratively merging pairs of locations, in order of distance, ensuring that the maximum intra cluster distance is less than Δt. The reported location of the cluster is defined to be the centroid of the locations contained within it.

4.3 Blacklisting

Certain types of road anomalies may produce high-energy, pothole-like events, yet may not represent a road surface in disrepair. This may include unusually shaped speed bumps, some types of bridges, and other equipment embedded in the road. For the occasional systematic error, we use a simple blacklist containing the locations of such road anomalies, and automatically remove detections in these locations from the final report. Blacklists for railroad-crossings, speed bumps and other well-known in-road equipment can be automatically generated from GIS data sources. For the purpose of evaluation, we did not use a blacklist.

4.4 A Note on False Negatives

There are several reasons why the absence of a detection at a particular location may not be indicative of smooth road. First, the error in GPS measurements (typically 5 meter median) is significantly larger than the size of the typical pothole. Thus, it is not possible to determine, purely by GPS localization, whether the wheel of a vehicle actually made contact with a given road anomaly. Second, typical drivers (those who are not collecting our training data) usually strive to *avoid* potholes, so the probability of hitting a road anomaly is likely to be considerably lower than what would be expected from an unbiased distribution based on road area.

Note that although this will tend to lower the rate at which we detect road surface anomalies, the process also biases our detections toward road surface anomalies that are difficult to avoid. Thus, those road surface anomalies that require the most urgent attention will tend to be detected sooner than those which can be safely avoided.

5. EVALUATION

The overall goal of this evaluation is to determine the accuracy and effectiveness with which P^2 is able to detect substandard road conditions. Evaluating P^2 presented us with some rather unique difficulties. First, detecting potholes based on their impact on the tires of a vehicle can be difficult even for a human passenger: there is a wide variance in the experienced impact from a single road surface anomaly between drives, primarily depending on exactly how the road surface anomaly was hit by the tire, and the speed of travel. Second, the goal of P^2 is to make use of existing moving vehicles to scan road surfaces around the city, but since we cannot be present in the cars as they drive, we cannot know what the experience in the car was at the time.

Due to these limitations, we kept the goals for our detector relatively modest:

- The detector must have a very low false negative rate for smooth road. Since most roads are in fact smooth road, flagging even a small percentage of smooth road as road anomalies would lead to an unacceptable number of misdetections.
- False negatives for potholes and other anomalies are a lesser concern; though we would obviously like to detect as many potholes as possible, missing a few potholes is OK.

In this section, we evaluate P^2 in four steps:

- **Classification accuracy on hand-labeled data.** Given the hand-labeled dataset described in §3.3, we divide it randomly into test and training sets. After training the detector on the training set, we determine the detection accuracy the test set. This procedure is repeated over many trials, and the average detector performance is reported.
- **Performance improvement using loosely labeled training data.** Using hand annotations about the overall quality of a road, we fine-tune the detector and determine the impact on detection accuracy on hand-labeled data.
- **Performance on loosely labeled roads.** Again, using the hand annotations about overall road quality, we compare the output of P^2 with the expected results. This allows us to characterize the false positive rate.
- **Spot-checks on uncontrolled data.** Finally, we manually spot-check each of the 48 highest-confidence detections, to verify that the road was indeed in poor condition.

5.1 Data Sets

We use three data sets in our evaluation:

- The carefully labeled data described in §3.3.
- The loosely labeled data described in §3.4.
- Finally, the vast majority of data was recorded through opportunistic mobility using the taxi P^2 deployment. We report data from over 10 days of driving, covering 9730 kilometers of road in the Boston metropolitan area.

We describe the performance of the detector on these three data sets in the next three sections.

5.2 Performance on Labeled Data

The purpose of this part of the evaluation is to study the baseline detection performance of P^2. To this end, we randomly split the labeled data into test and training sets, and computed the average detector performance on the test data after training the detector using the training data. Results here are the average of 100 such trials.

Class	before	after
Pothole	88.9%	92.4%
Manhole	0.3%	0.0%
Exp. Joint	2.7%	0.3%
Railroad Crossing	8.1%	7.3%

Table 3: Test data of listed class that was reported as potholes by our algorithm, before and after training on additional loosely labeled data.

Table 3 reports the performance of the detector on the labeled data, before and after training on additional "loosely labeled" data. Here, the percentages represent the fraction of reported potholes that were actually of the listed class. For example, 2.7% of reported pothole detections were actually generated by expansion joints before the additional training data. After the additional training data, these misdetections dropped to 0.3%. These reductions are obtained because we know that much of our loosely labeled data has no manholes or expansion joints. Note that in no case do we confuse smooth road for a pothole on our test data.

Some misdetections are to be expected. In particular, potholes can usually be differentiated from from railroad crossings, expansion joints, speed bumps and cross-walks, which usually create a simultaneous impact on two wheels, resulting in a smaller X acceleration component. However, these are sometimes uneven or at an angle with the road, confusing our detector. Using a GIS system, it should be possible to filter out these detections automatically, by rejecting potholes that are near railroad crossings or other features.

Judging from Table 3, it would appear that the false positive rate of P^2 is 7.6%. However, the labeled training data does not reflect the prior distribution of road anomalies over road segments. For example, 6% of training data samples are railroad crossings, yet only a tiny fraction of road segments contain railroad crossings.

Unfortunately, the prior distribution of road anomalies over road segments is not known. Below, we attempt to more accurately estimate the false positive rate using loosely labeled data.

5.3 Estimating the False-Positive Rate

Road	# potholes	#win	#det.	rate
Storrow Dr.	few	1865	3	0.16%
Memorial Dr.	few	1781	2	0.12%
Hwy I-93	few	2877	5	0.17%
Binney St	some	6887	25	0.63%
Beacham St	many	1643	231	14%

Table 4: Total number of detections over several passes of known roads.

To estimate the false positive rate of the detector, we run the detector on our set of loosely labeled traces. Table 4 summarizes the results of this experiment. For each road, we report the assessed condition of the road, the number of sample windows collected for the road, the number of reported detections, and the detection rate, being the number of detections over the total number of windows.

The first three roads are all in reasonably good shape, which is reflected in the low detection rate for these roads. Some of the detections on these roads may be legitimate, as even a good road may have the occasional bad spot. This means that the detection rate may be higher than the actual false positive rate. However, this establishes an upper bound on the false positive rate of at most 0.15% on good roads.

The last two roads in the list have a large number of real potholes. The results on these roads show a rough correspondence between the assessed road quality and the detection count.

5.4 Impact of Features and Thresholds

As discussed in §4, each sample window is passed through a series of filters before a detection is reported. In this section, we study the effect of each of the three main filters: peak-z, xy-ratio and speed/z-ratio. Figure 7 plots example numbers of "pothole detections" reported for each class of training data, as the z-peak threshold is varied. Here, the t_x and t_s values used were the ones that resulted in the best score (see §4.1) for some value of t_z.

Note that ideally, only the "pothole" class of training samples should result in pothole detections.

The three graphs show the detection behavior for a combination of 1, 2 and 3 filters respectively. We use the z-peak threshold t_z as an illustrative example; similar plots could be drawn for xz-peak and speed vs. z-peak thresholds, but were excluded here for brevity. In Figure 7(a), only the z-peak filter is employed. Here, expansion joints create a large z-peak, resulting in frequent misclassification of expansion joints as potholes. In Figure 7(b), the xz-ratio filter is added, which discards most expansion joints due to the lack of a large x impulse. However, to filter out all the expansion joints, the xy-ratio filter needs a very conservative threshold, $t_x = 1.5$, resulting in a small number of detections overall. Finally, Figure 7(c) adds the speed vs. z ratio filter. This raises z-peak threshold for high-speed events, discarding many high-speed expansion joint encounters. The addition of this new filter allows the xz-ratio filter to use a less conservative threshold, $t_x = 2.5$, without increasing the number of false positives. This recovers many of the potholes that were filtered out with the more conservative $t_x = 1.5$, which explains the increase in detections in figure 7(c).

Thus, with the three filters operating in concert, we achieve both a small number of mistakes, and a large number of correct detections. For our experiments on uncontrolled data, we use the parameter values $t_{z,x,s}$ that maximize the detector score on the training data.

5.5 Performance on Uncontrolled Cab Data

Kilometers of driving	9730
Kilometers of road covered	2492
Cars used	7
Total number of windows	1400000
Number of severe road surface detections	4131
Unique pothole segments after clustering	2709

Table 5: Total data collected from cab testbed.

In this section, we look at the ability of P^2 to detect anomalies in accelerometer recordings collected from taxis during normal service. This is a considerably larger data set, with roughly 400 million samples, or 250 hours of driving.

The goal of these experiments is to study the performance and accuracy of P^2 under completely uncontrolled conditions. Typical error sources under uncontrolled conditions include:

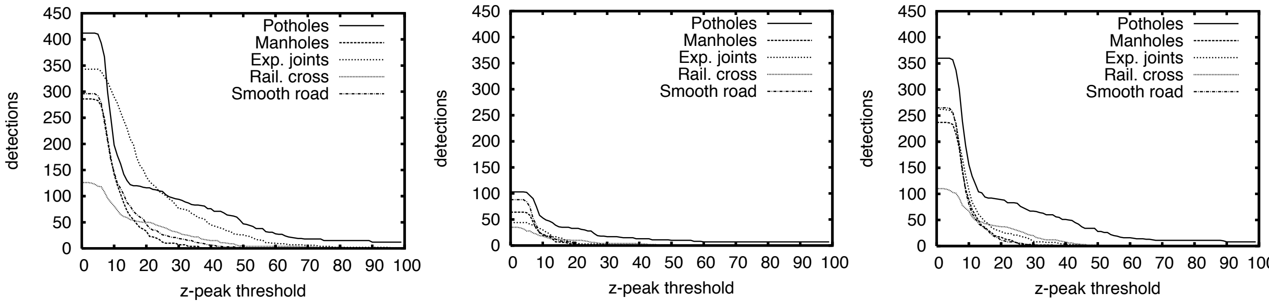

Figure 7: Pothole detections per class for varying z-peak threshold. (a) up to and including z-peak filter, (b) up to and including xy-ratio filter with $t_x = 1.5$ (c) up to and including speed vs. z ratio filter, with $t_x = 2.5$ and $t_s = 5$.

Figure 8: Number of clusters as a function of the cluster size threshold. 48 clusters with 4 or more members were found.

1. Drivers and passengers slamming doors as they enter or exit the vehicle,
2. Drivers and or passengers drumming on the dashboard, or opening the glove compartment,
3. Passengers hitting the lid of the glove compartment while adjusting legs or luggage,
4. Unusual driving behavior, such as swerving, braking hard, or accelerating rapidly, and finally
5. Drivers deliberately avoiding road surface anomalies.

After filtering out the majority of such events with signal processing, the remaining uncontrolled events are rejected using the clustering algorithm described in §4.2. We require that all detections occur in at least $k = 4$ traces before we report them as a detection. Otherwise, we assume the detection is a spurious event and do not report it. Figure 8 illustrates the rapid decrease in the number of valid detections as the minimum cluster size threshold is increased from 1 to 4.

This data produced road surface anomaly detections throughout the city. Table 5 summarizes the collected data, and the results of our clustering. Based on road surface anomaly detections, and the performance of our detector described in §5.2, we expect that a small fraction of these will not actually be potholes, with some fraction of such misclassifications being railroad crossings and other anomalies that can easily be eliminated without actually visiting the road.

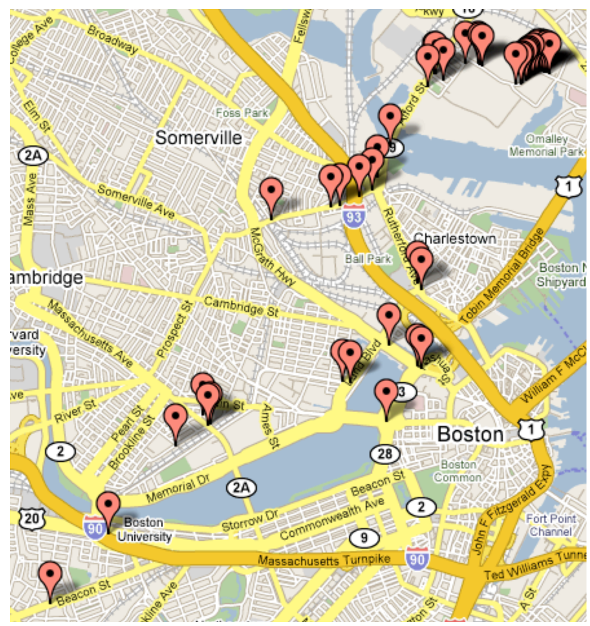

Figure 9: Partial map of Boston, and the 48 highest-confidence detections returned.

To verify the accuracy of P^2 on a wider range of streets and driving scenarios, and as a final quality control, we manually verified the set of 48 returned potholes where at least 4 independent detections were made. We visited each site, and determined what, if any class of road anomaly was present in the indicated location.

Figure 9 shows the set of detections on the map. The detections are spread throughout the Boston metropolitan area, but biased toward roads that the vehicles travel frequently, as this increases the probability of making at least 4 detections of a given pothole. The table below summarizes the results:

Reported detections $\geq 4\times$	48
Potholes	39
Sunk-in manholes	3
Railway crossings and exp. joints	4
Undetermined	2

Table 6: Spot-check results. The vast majority of detections represented road anomalies in need of repair. Some room for improvement remains, in particular in dealing with expansion joints on high-speed road segments.

Figure 10: Typical small pothole with missing pavement.

Figure 11: Severely sunk-in manhole, with eroded edges. This generated a pothole detection.

Here, potholes represent missing or severely sunk in and/or cracked pavement, as illustrated in Figure 10. Sunk-in manholes were badly enough sunk in and/or eroded to warrant repair, for example Figure 11. In the railway crossings and exp. joints category, one was a diagonal railway crossing, two were expansion joints on a curving flyover highway segment, and 1 was a large expansion joint on an old, high-speed bridge, shown in Figure 12. Finally, in the "undetermined" class, the two detections near the airport were difficult to diagnose, due to the hectic traffic situation and lack of any space to stop and look around. However, a cursory inspection (while driving) revealed no apparent potholes in these two locations.

Figure 12: Borderline expansion joint, on a high-speed bridge. This generated a pothole detection.

6. RELATED WORK

This section surveys related work on road surface monitoring and on data collection in sensor networks. P^2 is different from previous sensor networks in that it focuses on addressing the road surface monitoring problem in depth. The algorithms and sensing techniques we develop are also quite different from previous approaches to road surface assessment.

6.1 Road Surface Monitoring

Road conditions are a matter of public concern that have engendered a number of responses from local organizations dissatisfied with the state of their roads; solutions include: establishing "pothole hot-lines"[4], holding contests to report particularly bad potholes[5], and asking readers to contribute pictures of potholes[6]. We seek a more systematic approach to the problem, but hope that this public interest may cause volunteers to carry hardware in their cars.

Within the transportation technology industry, there has been some interest in road surface monitoring and in mapping and monitoring potholes. The current state-of-the-art in industry is to measure road surface quality using a "falling weight deflectometer"[7]. These devices apply a fixed load to the road surface and measure the distortion of the road to that load, giving an estimate of how imminent road surface failures are. Though these devices can give a reasonable measure of which roads are most in need of resurfacing, they are quite slow, as each measurement takes many seconds. According to the US Army Corps of Engineers [13], most states test fewer than 700 "lane-miles" of road per year using such devices (for reference, the New Jersey Turnpike is about 1200 lane-miles). Thus, we believe that P^2 could provide a valuable "first line of defense", alerting cities to areas that are most in need of testing and repair.

Within the transportation research community, the most commonly proposed approach to road surface monitoring and pothole detection involves the use of cameras. For example, Karuppuswamy et al. [12] describe a vision-based scheme for pothole avoidance in a mobile robot. They detect potholes by looking for

[4]ABC7Chicago.com: Operation Pothole: `http://abclocal.go.com/wls/story?section=traffic&id=3374965`

[5]Contest to Find Biggest Pothole: `http://www.wlns.com/Global/story.asp?S=6347658`

[6]Share your pothole photos online: `http://www.woodtv.com/Global/link.asp?L=228550`

[7]For example, see `http://www.dynatest.com/hardware/fwd_hwd.htm`

large (2 foot) circular objects in the field of view, which is clearly an idealization of reality. In general, using machine vision for this problem is unlikely to work well over the large range of road surface anomalies, and is complicated by the speed at which cars move as well as large variations in ambient lighting [14]. Recent work has proposed the use of ground-penetrating radar to perform similar detections [11]. These systems are fundamentally different from our approach of using sensor-equipped cars for opportunistic sensing, assessing road surface conditions over normal driving patterns and habits, rather than using special-purpose vehicles and equipment.

There have been a few proposals for the use of accelerometers to monitor road surfaces [1, 18], but these have been based on very small deployments and have largely focused on only demonstrating feasibility. For example, Angelini et al. [1] describe a single-accelerometer deployment in Worcester, MA where they use a data-logger to record the location of a few potholes. They do not provide a systematic categorization of potholes in the whole city, or describe methods for remote retrieval of sensor data.

Finally, there has been some work on classifying pothole severity and cataloging the types and causes of potholes. For example, Eaton et al. [6] corroborate our findings that potholes often arise around manholes, railroad tracks, expansion joints, and in intersections. They note that embedded metal objects cause roadway wear as paving around them tends to be thinner and more subject to decay; intersections receive more wear as cars stop and start as they pass through.

TrafficSense is a recent project from Microsoft Research[8] that shares with the CarTel project and the P^2 system the high-level idea of using mobile nodes to sense traffic and road conditions. Rather than using embedded nodes, TrafficSense uses GPS-enabled smartphones carried by drivers and triggered sensing methods to save energy. To gather acceleration samples without requiring careful placement, they map observed samples to known gravity, and to any transient decelerations caused by braking (tracked by the GPS). P^2 could incorporate those techniques, while TrafficSense could use the techniques developed in this paper to assess road conditions.

6.2 Data Collection in Sensor Networks

The P^2 software infrastructure is a general one for intermittently connected and mobile sensor networks. The closest related work in this area includes recent efforts on mobile sensor networks such as CarTel [10], Mobeyes [15], and participatory sensing [2]. Like CarTel, P^2 uses a delay-tolerant network stack. Unlike CarTel's declarative query system (ICEDB), however, P^2 uses a lightweight delay-tolerant pipe abstraction (dPIPE) that allows software components to be connected in a work-flow (modeled after how UNIX pipes work, extended to handle intermittently connected, distributed components). This approach allows components written in different languages (e.g., signal processing in WaveScript, data analysis in Perl, etc.) to be composed, and enables components that reduce the amount of data to be processed to run "in the net" on the cars. dPIPE is a simple abstraction, which should be easy to use by anyone familiar with UNIX pipes. Such approaches are also common in the pervasive computing community [3, 5], where they are used to provide service discovery and composability, sometimes in environments where users are mobile or intermittently disconnected.

There have been a number of proposals for systems to simplify the collection and processing of data from sensor network systems. Many of these take the form of "macroprogramming" languages, like TinyDB [16], Regions [19], Regiment [17], and Semantic Streams [20]. These approaches provide a high level language and a compiler that attempts to place computation in a network efficient way. Other approaches allow programmers to express their programs as data flow-like "tasks" that may be spread across several different nodes in the network [7, 8, 9, 4]. P^2 shares the same motivation—hiding complexities of distribution communication and coordination in the sensor network domain—as these high level application toolkits. The P^2 architecture is different from these previous systems, however, in that it emphasizes mobility and intermittent connectivity as key problems that must be handled by the architecture.

[8] `http://research.microsoft.com/research/mns/projects/TrafficSense/`

7. CONCLUSION

This paper studied an application of mobile sensing: detecting and reporting the surface conditions of roads. We described the P^2 system and associated algorithms to monitor this important civil infrastructure using a collection of sensor-equipped vehicles. P^2 uses the inherent mobility of the participating vehicles, opportunistically gathering data from vibration and GPS sensors, and processing the data to assess road surface conditions. We deployed P^2 on 7 taxis running in the Boston area. We use a signal processing and machine-learning based approach, and show that P^2 is well suited to detecting adverse road conditions. Via careful selection of training data and features, the P^2 detector misidentifies road features as having potholes less than 0.2% of the time in controlled experiments. We also evaluated our system on data from thousands of kilometers of "uncontrolled" taxi drives, and found that out of reported detections, over 90% contain road anomalies in need of repair.

Acknowledgments

We thank Christine Julien and the anonymous reviewers for useful comments that improved this paper. This work was supported by the National Science Foundation under grants CNS-0720079, CNS-0205445, CNS-0520032, and CNS-0509261.

8. REFERENCES

[1] N. Angelini, J. Brache, M. Gdula, and G. Shevlin. Gps coordinate pothole mapping. Technical report, Worchester Polytechnic, 2006. `http://users.wpi.edu/~sageman/mqp/docs/executive-summary.doc`.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory Sensing. In *World Sensor Web Workshop*, 2006.

[3] G. Chen, M. Li, and D. Kotz. Design and implementation of a large-scale context fusion network. In *Mobiquitous*, 2004.

[4] D. Chu, K. Lin, A. Linares, G. Nguyen, and J. M. Hellerstein. SDLIB: a sensor network data and communications library for rapid and robust application development. In *IPSN '06*, 2006.

[5] H. Cohen, H. Lei, P. Castro, J. S. D. II, and A. Purakayastha. Composing pervasive data using iQL. In *WMCSA*, 2002.

[6] R. Eaton, R. Joubert, and E. Wright. Pothole primer: A public administrator's guide to understanding and managing the pothole problem. Technical Report 81-21, US Army Corps of Engineers, Cold Regions Research & Engineering Laboratory, 1989. `http://www.crrel.usace.army.mil/techpub/CRREL_Reports/reports/SR81-21.pdf`.

[7] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *SenSys*, 2006.

[8] B. Greenstein, E. Kohler, and D. Estrin. A sensor network application construction kit (SNACK). In *SenSys*, pages 69–80, 2004.

[9] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin. Capturing high-frequency phenomena using a bandwidth-limited sensor network. In *SenSys*, pages 279–292, 2006.

[10] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *Proc. ACM SenSys*, Nov. 2006.

[11] D. R. Huston, N. V. Pelczarski, B. Esser, and K. R. Maser. Damage detection in roadways with ground penetrating radar. In *International Conference on Ground Penetrating Radar*, volume 4084, pages 91–94, Apr. 2000.

[12] J. Karuppuswamy, V. Selvaraj, M. M. Ganesh, and E. L. Hall. Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment. *Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, 4197(1):70–80, 2000.

[13] M. Kestler. Current and proposed practices for nondestructive highway pavement testing. Technical Report 97-28, US Army Corps of Engineers, Cold Regions Research & Engineering Laboratory, 1997. `http://www.crrel.usace.army.mil/techpub/CRREL_Reports/reports/SR97_28.pdf`.

[14] D. Kil and F. Shin. Automatic road-distress classification and identification using acombination of hierarchical classifiers and expert systems-subimage andobject processing. In *International Conference on Image Processing*, 1997.

[15] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. MobEyes: Smart Mobs for Urban Monitoring with a Vehicular Sensor Network. *IEEE Wireless Communications*, 2006.

[16] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. In *OSDI*, 2002.

[17] R. Newton, G. Morrisett, and M. Welsh. The regiment macroprogramming system. In *IPSN*, 2007.

[18] V. Rouillard. Remote monitoring of vehicle shock and vibrations. *Packaging Technology and Science*, 15(2):83–92, 2002.

[19] M. Welsh and G. Mainland. Programming sensor networks using abstract regions. In *NSDI*, 2004.

[20] K. Whitehouse, F. Zhao, and J. Liu. Semantic streams: A framework for composable semantic interpretation of sensor data. In *EWSN*, 2006.