Development of Micro Wireless Sensor Platforms for Collecting Data of Passenger-Freight Interactions

Mohammad Mozumdar

Graduate Students: Kyle Ying, Alireza Ameri, Ankit Trivedi, Dilip Ravindra, Darshan Patel, Varun Sharma, Ashutosh Parhad

Department of Electrical Engineering
California State University Long Beach
The thing we don’t like but have to face everyday – Road Traffic!
But we can make smarter turn with real-time traffic!

Google Maps showing optimized route from USC to Orange County
How exactly does Google Maps/Garmin/TomTom know how clogged the highway is on your way out to home or office?

The traffic information comes from a variety of sources:
• Commercial traffic data providers (INRIX, Tele Atlas, HERE, ..)
• Departments of Transportation
• State agency – Caltrans

Raw data is collected from:
• Mobile users (Google Maps)
• **Road sensors**
• Traffic cameras, and even through aircraft

This information is compiled and delivered via radio frequency (FM/HD Radio™ or satellite) to your navigation system.
Road Sensors: Inductive Loop

- Existing traffic/vehicle detection is determined with “Inductive Loop” technologies
- These loops generate a magnetic field that operates at frequencies typically less than 1kHz
- Large rectangular loops (4’ x 8’, 6’ x 8’, 6’x 12’) are used to detect larger vehicles
- Small size loops (i.e. 2’ x 5’, 3’ x 6’, 6’ x6’) are used to detect smaller vehicles, such as motorcycles and automobiles

Source: US DoT Federal Highway Administration
Road Sensors: Inductive Loop

(a) Magnetic anomaly induced in the Earth’s magnetic field by a magnetic dipole.

(b) Perturbation of Earth’s magnetic field by a ferrous metal vehicle.

Source: US DoT Federal Highway Administration
Inductive Loop Pros & Cons

Advantages
• Detects ferrous objects precisely
• Typically immune from environmental effects such as weather, temperature, a terrain variations

Disadvantages
• Expensive to install and maintain ($$$$
• Relatively significant power usage for the generation of the magnetic field.
• Large area usage (greater than 10 sq.ft.)
Proposed Solution For Smart Road

- Magneto Resistive Sensor (HMC5883L)
- Lithium battery/Energy Scavenging
- Microcontroller with RF Core (CC430F5137)

Simplified Hardware Architecture of Embedded Wireless Sensor Platforms (will be implanted in road)

Tier 1 networking between Sensors and ECU

Smart Highway

Sensor Node

Wireless Communication between sensors and ECU (RF Transceiver)

TDMA Based wireless STAR network

Electronics Communication Unit (ECU)

Solar Cell

GPRS Data Communication

Tier 2 networking between ECU and Data Center

County/Regional Traffic Data Storage & Control Center

RF Transceiver (CC 2420)

GPRS Shield (SIM900)

Single Board Computer (Raspberry Pi)

Simplified Hardware Architecture of ECU Platforms (Pull Box)
Proposed VS Traditional Inductive Loop Based System

Proposed Wireless MR Sensor Based Approach

- Sensor Node (Credit Card Size)
- Wireless Communication

Electronics Communication Unit (ECU)

Solar Cell

Traditional Inductive Loop Detector Based Approach

- 6ft x 6ft
- 6ft x 6ft
- 6ft x 6ft

Pull Box

Control Box

AC power supply
Anisotropic Mangeto-Resistive Sensors (AMR) IC Sensor

**AMR Sensor IC** (Honeywell HMC5883L 3-axis magnetometer - 3mm in size)
- Wheatstone bridge variable resistor network that changes resistance w.r.t. changes to the magnetic field
- Provides the same advantages to inductive loop technologies without the power and area disadvantages
- Power consumption extremely low (~200μA at lower sampling rates)

**Microcontroller** (CC430F5137)
- Low power modes (LPM) for sleep between computational and communication operations
- Single package μproc and RF core for low area wireless transmissions

Sources:
- Honeywell
- Texas Instruments
Machine Learning Based Vehicle Classification

• Useful when the sets of data is large enough that human observations for extracting patterns in data become impractical.
• Typically associated with the field of data mining
• Pattern recognition based on a set of rules

General Idea:
• Collect vehicle data crossing the AMR sensor
• Utilize ML tools to generate a model for classification
Our Lab Testbed Setup

- 7 different RC Vehicles with a variety of similar and different attributes
- 7 ft straight track for each vehicle to make passes
- 2 sensors roughly 4’ apart to take gather readings and classify
Data Collection for supervised machine learning

• We collected data for each of the 7 vehicles across 350 runs over 2 sensors
• Total: 700 samples, 100/class for training

Why a decision-tree based algorithm?
• Simple and computationally efficient tree
• Simplicity of implementation in software
Implementation Flowchart

1. Normalize Raw Data
2. Vehicle Threshold Detection
3. Record Data in Detection Window
4. Classify Vehicle using Decision Tree Model
5. Feature Extraction at Window End
6. Output Vehicle Classification
Adaptive Baseline

- Zeroing the background environmental magnetic field by offset
- Allows for the reuse of the same vehicle detection and classification algorithm in multiple environments
- Noise removal can be implemented at this stage
Threshold Detection

- Once a vehicle passes the threshold the detection flag triggers and a certain number of samples are recorded for processing.
Features Collected from Vehicles

Interesting Features:

- **Min**: minimum value of an axis during the detection window
- **Max**: maximum value of an axis in the window
- **Mean**: average of all axis values in the window
- **Range**: Maximum – Minimum

Using a 3-axis sensor this results in 12 unique features

These Features are very simple to calculate and compute
Example Plot and Feature Extraction

MinX=-1097
MinY=-256
MinZ=-834
MaxX=1054
MaxY=267
MaxZ=1011
MeanX=4.01
MeanY=17.13
MeanZ=7.48
RangeX=2151
RangeY=523
RangeZ=1845
Example Car Data

Similar Sizes but Different Signatures!
Machine Learning: decision tree learning (J48)

J48 is the open source Java implementation of C4.5/ID3 developed by John Quinlan

Inputs: multiple features corresponding to a single classifier
Note: higher # samples per classifier results in a more accurate output tree

Output: a decision tree with the highest classification rate given the features

Source: University of Waikato, WEKA
WEKA output

J48 pruned tree

4 Features Selected

\[
\text{range}_z \leq 2187 \\
\quad \text{max}_x \leq 419.4: 1 \ (100.0) \\
\quad \text{max}_x > 419.4 \\
\quad \quad \text{max}_x \leq 791.6 \\
\quad \quad \quad \text{max}_x \leq 757.8: 5 \ (94.0) \\
\quad \quad \quad \text{max}_x > 757.8 \\
\quad \quad \quad \quad \text{min}_y \leq -118.2: 2 \ (4.0) \\
\quad \quad \quad \quad \text{min}_y > -118.2: 5 \ (6.0) \\
\quad \quad \quad \text{max}_x > 791.6: 2 \ (96.0) \\
\quad \text{range}_z > 2187 \\
\quad \quad \text{min}_y \leq -1103.8 \\
\quad \quad \quad \text{range}_z \leq 3725: 6 \ (60.0) \\
\quad \quad \quad \text{range}_z > 3725 \\
\quad \quad \quad \quad \text{mean}_z \leq -168.58: 4 \ (101.0/1.0) \\
\quad \quad \quad \quad \text{mean}_z > -168.58: 6 \ (36.0) \\
\quad \quad \quad \text{min}_y > -1103.8 \\
\quad \quad \quad \quad \text{mean}_z \leq -116.47 \\
\quad \quad \quad \quad \quad \text{range}_z \leq 3796: 6 \ (3.0) \\
\quad \quad \quad \quad \quad \text{range}_z > 3796: 7 \ (100.0) \\
\quad \quad \quad \quad \quad \text{mean}_z > -116.47: 3 \ (100.0)
\]

Number of Leaves : 11
Size of the tree : 21
Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===

--- Summary ---

Correctly Classified Instances 692 98.8571 %
Incorrectly Classified Instances 8 1.1429 %
Kappa statistic 0.9867
Mean absolute error 0.0037
Root mean squared error 0.0572
Relative absolute error 1.4927 %
Root relative squared error 16.3407 %
Total Number of Instances 700

=== Confusion Matrix ===

\[
\begin{array}{cccccccc}
\text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} & \text{g} & \text{--- classified as} \\
100 & 0 & 0 & 0 & 0 & 0 & 0 & a = 1 \\
0 & 99 & 0 & 0 & 1 & 0 & 0 & b = 2 \\
0 & 0 & 100 & 0 & 0 & 0 & 0 & c = 3 \\
0 & 0 & 0 & 97 & 0 & 3 & 0 & d = 4 \\
0 & 1 & 1 & 0 & 98 & 0 & 0 & e = 5 \\
0 & 0 & 0 & 1 & 0 & 99 & 0 & f = 6 \\
0 & 0 & 1 & 0 & 0 & 0 & 99 & g = 7 \\
\end{array}
\]
Graphical Tree

<table>
<thead>
<tr>
<th>Node Num.</th>
<th>Min Y</th>
<th>Max X</th>
<th>Mean Z</th>
<th>Range Z</th>
<th>Best Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9527</td>
<td>0.8002</td>
<td>0.9031</td>
<td>0.9868</td>
<td>Range Z</td>
</tr>
<tr>
<td>2</td>
<td>0.3339</td>
<td>0.9707</td>
<td>0.3978</td>
<td>0.3846</td>
<td>Max X</td>
</tr>
<tr>
<td>3</td>
<td>0.3060</td>
<td>0.8411</td>
<td>0.2155</td>
<td>0.3068</td>
<td>Max X</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.1735</td>
<td>0</td>
<td>0.0379</td>
<td>Max X</td>
</tr>
<tr>
<td>5</td>
<td>0.7108</td>
<td>0</td>
<td>0.7108</td>
<td>0.0570</td>
<td>Min Y</td>
</tr>
<tr>
<td>6</td>
<td>0.9317</td>
<td>0.1864</td>
<td>0.7574</td>
<td>0.4976</td>
<td>Min Y</td>
</tr>
<tr>
<td>7</td>
<td>0.2047</td>
<td>0.1601</td>
<td>0.3717</td>
<td>0.4249</td>
<td>Range Z</td>
</tr>
<tr>
<td>8</td>
<td>0.1386</td>
<td>0.4467</td>
<td>0.8798</td>
<td>0.0084</td>
<td>Mean Z</td>
</tr>
<tr>
<td>9</td>
<td>0.5724</td>
<td>0.3189</td>
<td>0.9625</td>
<td>0.8676</td>
<td>Mean Z</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0.0066</td>
<td>0.6627</td>
<td>Range Z</td>
</tr>
</tbody>
</table>

- **N1**: Range Z
- **N2**: Max X
- **N3**: Max X
- **N4**: Max X
- **N5**: Min Y
- **N6**: Min Y
- **N7**: Range Z
- **N8**: Mean Z
- **N9**: Mean Z
- **N10**: Range Z

- **Yellow**
- **Red**
- **BigGreen**
- **Big Car**
- **Diecast**
- **Buggy**
Brute Force Search for Best Results

• The output tree doesn’t always generate the best results given a large number of features
• Due to fast processing time to generate the output tree, we can easily calculate all combinations (n choose k)
  \[
  \binom{n}{k} \text{ or } nC_k
  \]
• We use \( n=2,3,4 \) where \( k=12 \)
# Feature Performance

## 2 features (66 combinations):

<table>
<thead>
<tr>
<th>Comb#</th>
<th>Classification%</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>94%</td>
<td>minx maxx</td>
</tr>
<tr>
<td>58</td>
<td>93%</td>
<td>minx rangex</td>
</tr>
<tr>
<td>30</td>
<td>93%</td>
<td>maxx rangey</td>
</tr>
<tr>
<td>60</td>
<td>91%</td>
<td>minx meany</td>
</tr>
</tbody>
</table>

## 3 features (220 combinations):

<table>
<thead>
<tr>
<th>Comb#</th>
<th>Classification%</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>219</td>
<td>98%</td>
<td>minx miny maxx</td>
</tr>
<tr>
<td>200</td>
<td>98%</td>
<td>minx maxx maxz</td>
</tr>
<tr>
<td>194</td>
<td>97.8571%</td>
<td>minx maxx rangez</td>
</tr>
<tr>
<td>57</td>
<td>97.8571%</td>
<td>maxx rangey rangez</td>
</tr>
<tr>
<td>149</td>
<td>97.7143%</td>
<td>miny maxx rangez</td>
</tr>
</tbody>
</table>

## 4 features (495 combinations):

<table>
<thead>
<tr>
<th>Comb#</th>
<th>Classification%</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>479</td>
<td>98.8571%</td>
<td>minx miny maxx rangez</td>
</tr>
<tr>
<td>270</td>
<td>98.8571%</td>
<td>miny maxx meanz rangez</td>
</tr>
<tr>
<td>390</td>
<td>98.7143%</td>
<td>minx maxx meanz rangez</td>
</tr>
<tr>
<td>78</td>
<td>98.7143%</td>
<td>maxx meany meanz rangez</td>
</tr>
</tbody>
</table>
Best Results Simulated vs. Testbed

Simulated Results

<table>
<thead>
<tr>
<th>Cross-Validation Percentages</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three Features (maxx, rangey, rangez)</td>
<td>97.86%</td>
</tr>
<tr>
<td>Three Features (miny, maxx, rangez)</td>
<td>97.71%</td>
</tr>
<tr>
<td>Four Features (minx, miny, maxx, rangez)</td>
<td>98.86%</td>
</tr>
<tr>
<td>Four Features (miny, maxx, meanz, rangez)</td>
<td>98.86%</td>
</tr>
</tbody>
</table>

Actual Results:

<table>
<thead>
<tr>
<th>Real-world Classification Percentages</th>
<th>Number of Features (Attributes)</th>
<th>Real-world</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Three Features (maxx, rangey, rangez)</td>
<td>98.57%</td>
</tr>
<tr>
<td></td>
<td>Three Features (miny, maxx, rangez)</td>
<td>97.38%</td>
</tr>
<tr>
<td></td>
<td>Four Features (minx, miny, maxx, rangez)</td>
<td>90.24%</td>
</tr>
<tr>
<td></td>
<td>Four Features (miny, maxx, meanz, rangez)</td>
<td>99.05%</td>
</tr>
</tbody>
</table>

Simulated results match real world testing values very closely.

Note: minx results are lower due to clipping
Energy Scavenging Using Piezoelectric Sensors

• Mechanical to Electrical energy conversion
• Proper implementation can help in continuous operation of wireless sensors
• Almost 70% of the overall efficiency of the energy scavenging system depends on Piezoelectric sensors
• Applications include consumer electronics, automotive, health, WSN, etc.

Pressure generated by tires of cars on the piezoelectric sensors

Generated power is stored in batteries
Energy Scavenging System

- Energy Source
- Piezoelectric Sensors
- Mechanical Energy to Electrical Energy
- Rectification Circuit
- ADC
- Voltage Regulation
- Charge Regulation Circuit
- Switch
- Rechargeable Battery
- Power Storage Element

The system converts mechanical energy into electrical energy using piezoelectric sensors, which are then rectified and regulated before being stored in a rechargeable battery.
Lab Prototype of Energy Scavenging System

- Wooden frame
- Padding to support wooden frame
- Adhesive padding to support sensors
- Piezoelectric sensors
- Internal wire connections
- External wire connections
- Screws

Side View of hardware - arrangement of piezoelectric sensors

Hardware - Internal wire connections

Top view of hardware - arrangement of piezoelectric sensors
Advanced Energy Scavenging System

- 3 layers instead of 1 layer/ Smaller size and less implementation cost
- Increase probability of the sensors being pressed in every tap
- Increase number of sensors being pressed in a single tap
Energy Scavenging System

- 1 AA rechargeable battery can be charged in 10-12 hours with vehicles and pedestrians passing over the sensors in every 5 seconds using the designed hardware.
- The sensors placed on crosswalks can increase the average number of taps.
- Charging rate would be better if the efficiency and the number of sensors used are increased.
Designed Smart Traffic Sensing Node

- Size: 2 ½” x 1 ½” x ¾” (with AA battery pack)
- Dimensions will change depending on the battery pack used in future implementations
Final Remarks

The system described in this presentation can replace current inductive loop technologies with:

– Maintain traffic/vehicle detection capabilities
– Additional features such as vehicle classification
– Lower power consumption
– Lower physical area utilization

In addition, many classifiers can be used at high accuracy rates compared to other methods utilizing solely novel features.
Related Publications


Questions?