# Wolverine: Traffic and Road Condition Estimation using Smartphone Sensors

Ravi Bhoraskar, Nagamanoj Vankadhara, Bhaskaran Raman, Purushottam Kulkarni
Indian Institute of Technology, Bombay

## Introduction

- Growing population : Growing Number of vehicle users
- Growing vehicular users : Growing traffic
- Need a mechanism to estimate traffic
  - Avoid congested roads
  - Find potholes remotely
- Smartphones becoming ubiquitous - Android phones available for Rs. 5000

## Problem Statement

- Design a smart phone based solution
- Traffic estimation: free flowing vs congested: use of braking detection
- Road conditions and anomalies: smooth vs bumpy: bumps and potholes

## Related Work

| Method | Interesed in | Hardware | Scalability | Accuracy |
|---|---|---|---|---|
| Auto Witness [Sensys '10] | Vehicle Trajectory | Accelerometer, Gyro, GSM | No | >90% |
| Pothole Patrol [MobiSys '08] | Road State detection | Accelerometer, GPS | No | < 0.2% false positives |
| Road Sound Sense [Secon '11] | Vehicle speed | Acoustic sensors | No | accuracy varying b/w 85.7% to 100% |
| Nericell [Sensys '08] | Vehicle acceleration | Accelerometer, Microphone, GPS, GSM | Yes | 11.1% false positives, 22% false negatives |
| Wolverine (Our Method) | Vehicle acceleration | Accelerometer, Magnetometer, GPS | Yes | - |

Interested in a scalable mobile based solution, with low deployment cost.

## Need for reorientation

- Phone can be placed arbitrarily with respect to the vehicle
- Event detection algorithms work on acceleration from X, Y and Z axes differently (detect braking as bump in vertical phone)
- Phone orientation can change even during vehicle motion

## Reorientation Framework

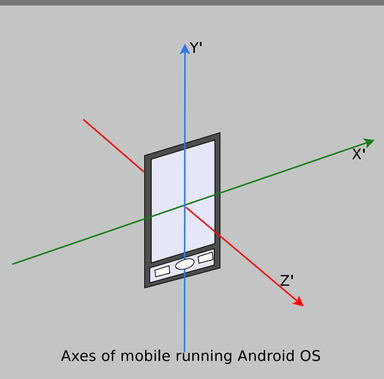



Figure: Phone Axes

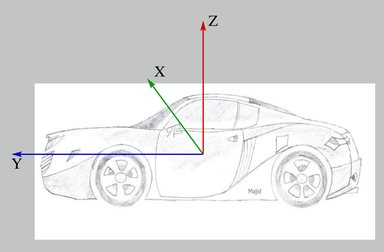



Figure: Vehicle Axes

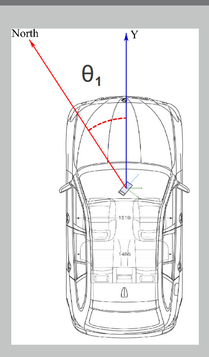



Figure: Vehicle Bearing

## Virtual Reorientation Algorithm

- Reorientation in Nericell
  1. Wait till angle with X-Y plane changes, to trigger reorientation
  2. Use accelerometer to compute angle with X-Y plane
  3. Turn on GPS, and wait for braking event. Use $\vec{a_Y} = \vec{a} - \vec{a_Z}$ to compute Y
- Reorientation in Wolverine
  1. Find angle with X-Y plane, like Nericell
  2. Find angle with north, in phone coordinates
  3. Calculate direction of motion using GPS
  4. Find angle with north, in vehicle coordinates
  5. Subtract the vectors to find the bearing of phone w.r.t. vehicle

## Energy Consumption in Reorientation

- GPS is major energy consumer. GPS On Time:
  - Nericell: TTTF + Time till Braking + Time for Braking
  - Wolverine: (TTTF + Time to Record) × 2

| Modality | Power | Time(Nericell) | Time(Wolverine) |
|---|---|---|---|
| GPS | 617.3mW | 10% | 1.1% |
| Sensors + CPU | 31.85mW | 100% | 100% |

- Energy Savings compared to Nericell
  - Per reorientation Event : 89%
  - Total : 58%

## Machine Learning Algorithms: Motivation

- Nericell uses fixed thresholds on accelerometer values
- Threshold may vary across vehicles, road conditions and the mobile device
- Let the thresholds be learned, for better performance

## Machine Learning Algorithm: Technique

1. First, reorient the accelerometer data
2. Divide into 1 second windows
3. Extract features from each window
4. Use k-means on the training data, then label it
5. Use labeled data to train SVM
6. Classify the incoming data using SVM

- Features Used
  - Bump Detection: $\sigma_Z$ - Standard Deviation in **Z** Direction
  - Braking Detection: $\delta_Y$ - Amplitude in **Y** Direction

## Experimental Setup

- Used **HTC Wildfire S** and **Samsung Nexus S**
- Both running **Android OS 2.3.3** (Gingerbread) and SDK Version 10
- Both have **accelerometer, magnetometer** and **GPS sensors**
- Nexus S has **gyroscope** as well (did not use this for now)
- Collected data on *Suzuki Access 125* and *Bajaj Autorickshaw* in IIT-Bombay campus

## Bump Detection

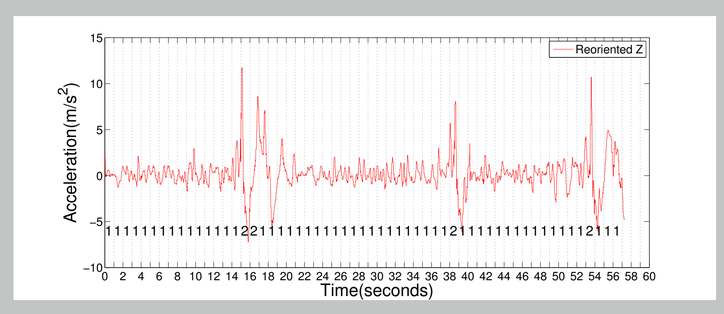



Figure: Training

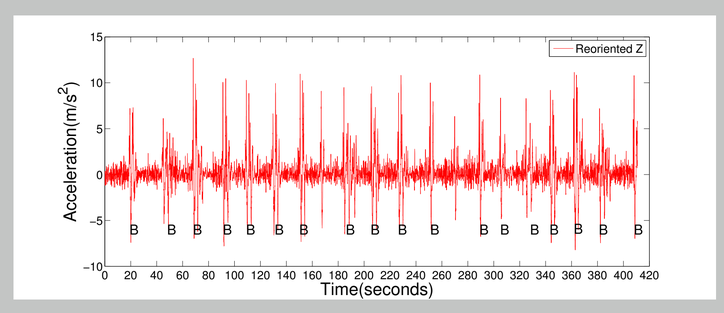



Figure: Testing

Brake detection not shown due to lack of space

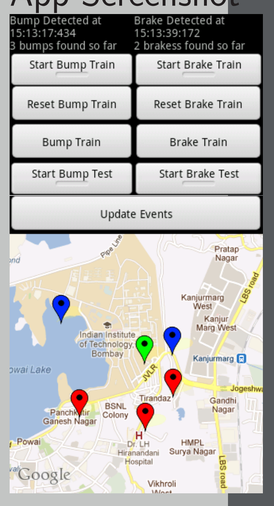
## Results

| | False Positives | False Negatives |
|---|---|---|
| Bump | 0% | 10% |
| Braking | 2.7% | 21.6% |

## Conclusions and Future Work

- Conclusions
  - Traffic state estimation is possible, by braking detection
  - Road state estimation is poosible, by bump detection
  - Scalable system, as any user having smartphone can participate
  - Use of magnetometer reduces the energy in reorientation of accelerometer axes
- Future Work
  - Fully implement application that can be installed by the users in their smartphones, with map annotation
  - Reduce energy consumption, by detecting other users in proximity
  - Record energy consumption for a better energy model
  - Localization in energy efficient manner
  - Differentiating class of vehicles

App Screenshot






http://www.cse.iitb.ac.in {ravi, nagamanojv, br, puru}@cse.iitb.ac.in