Presented by Chuong Ngo

Using Mobile Phone Barometer for Low-Power Transportation Context Detection

By Kartik Sankaran, Minhui Zhu, Xiang Fa Guo, Akkihebbal L. Ananda, Mun Choon Chan, and Li-Shiuan Peh

Smart in Smartphones

- Intelligent behavior
  - Context aware apps
- Sensors = context
  - Increased power consumption
  - Specialized hardware

Transportation-mode Detection

- Useful for many things
  - Personal tracking
  - Regional data/planning
- Usually accelerometers
  - Low power
  - 10+ Hz sampling
  - Complex ML model

Is there a better way?

A Flash

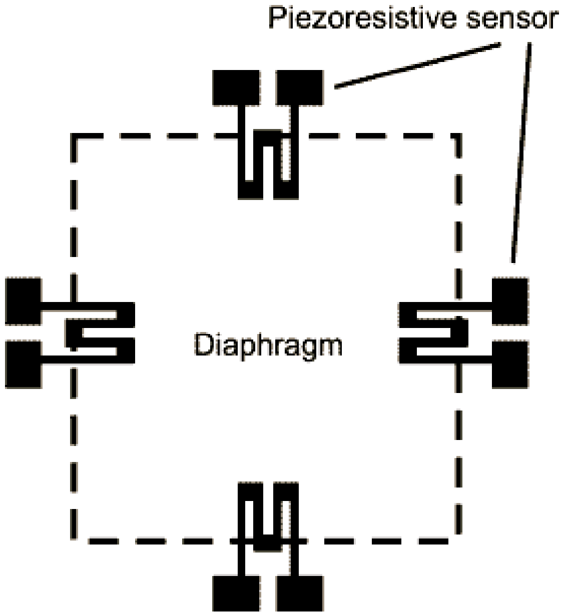
- Terrain is not perfectly flat
- Faster transportation = faster traversal of terrain
- More vertical movement for faster modes of transporation

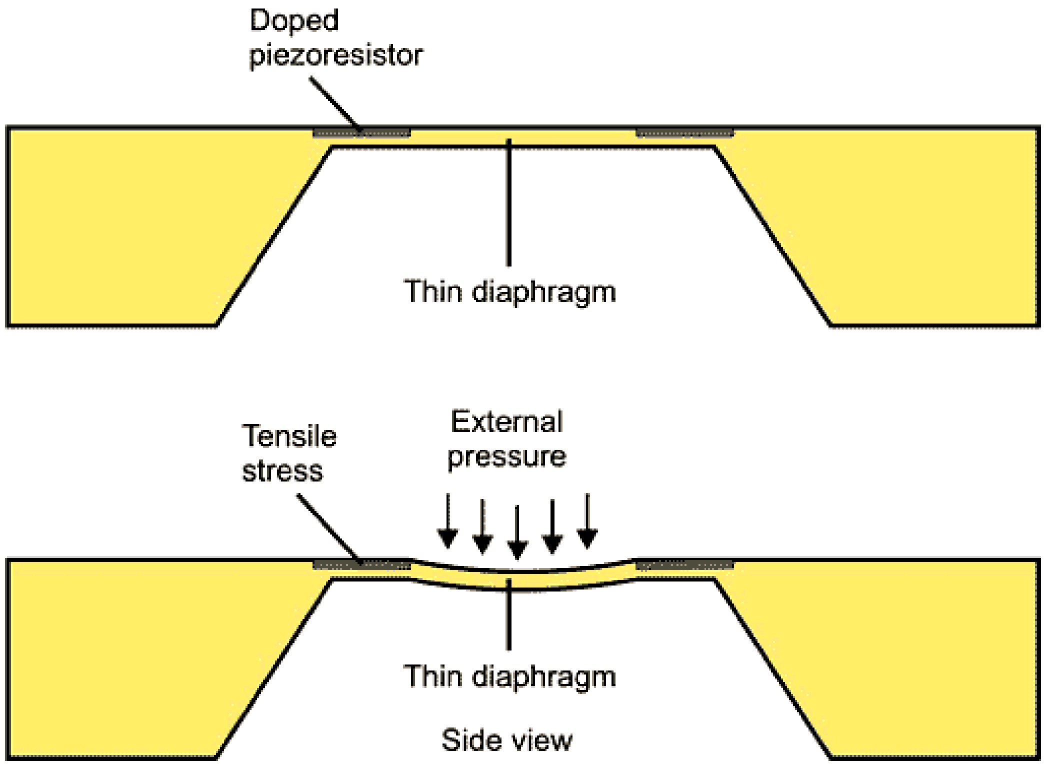
Barometer Background

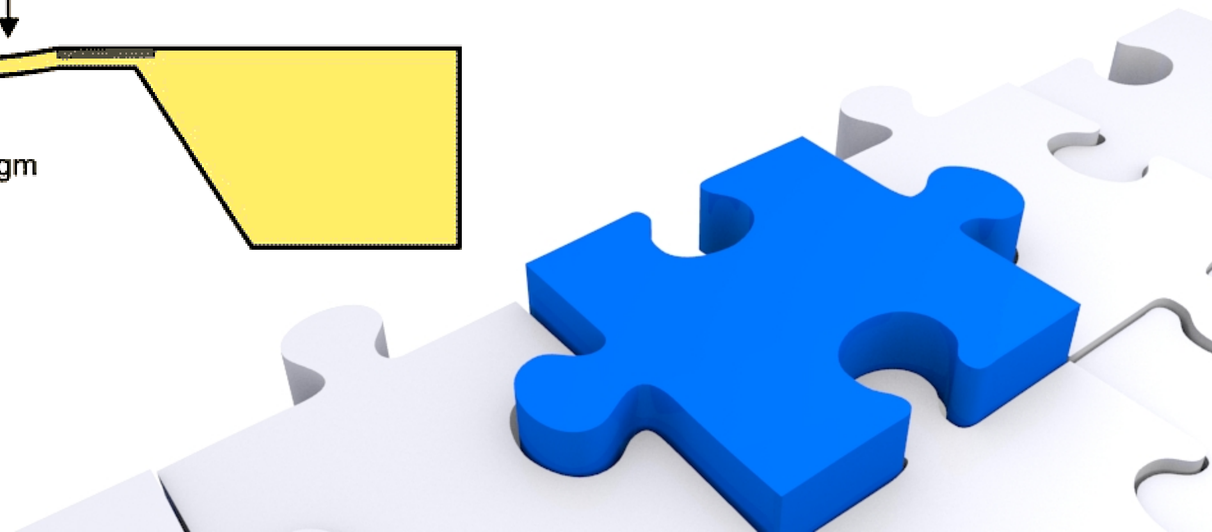
- Orientation and position independent
- Simpler calibration
- Better WAIT detection
- Lower necessary sampling rate

- Barometric pressure unstable
  - Altitude
  - Weather
  - Temperature
- Installation bias
- Aging drift

Barometer Breakdown

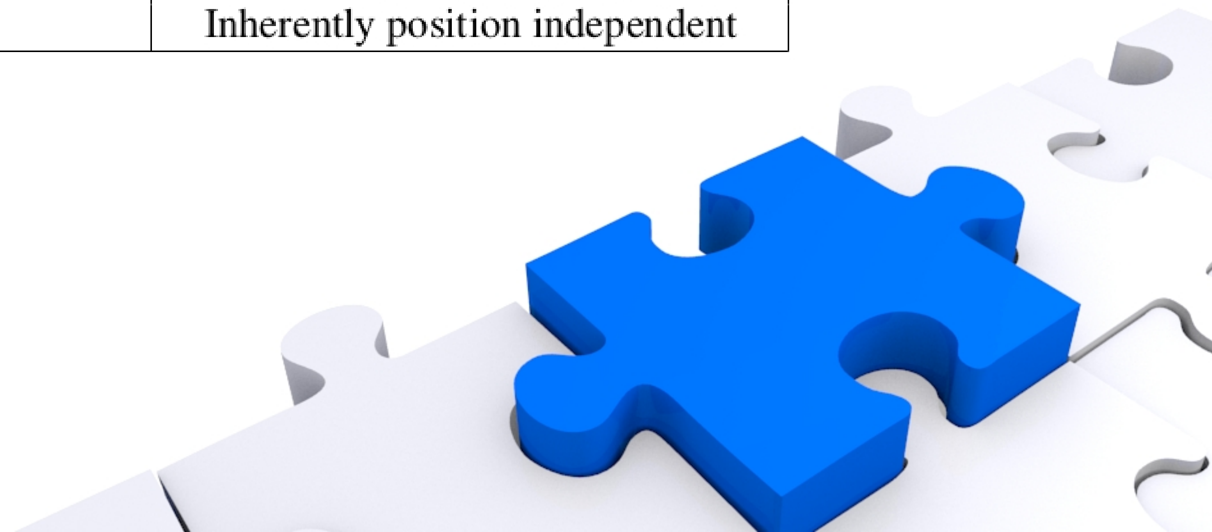






Methodology Comparisons

Sensor	Power	Limitations	Barometer advantage
GPS	Very high	Lack of indoor/underground coverage High power usage	Usable everywhere Ultra-low-power
WiFi/Cellular	High/Moderate	Requires dense access points/cellular towers	No external infrastructure
Accelerometer	Low	Extensive training required Classification complexity Position dependence	Simple calibration based on terrain Simple processing Inherently position independent



Power Play

	Power (mW)	Increase over base power
Accl (20 Hz)	230	112%
Accl (10 Hz)	180	67%
Accl (2 Hz)	164	51%
Baro (1 Hz)	110	2%
CPU Awake (base)	108	x
CPU Asleep	25	x

Power Play

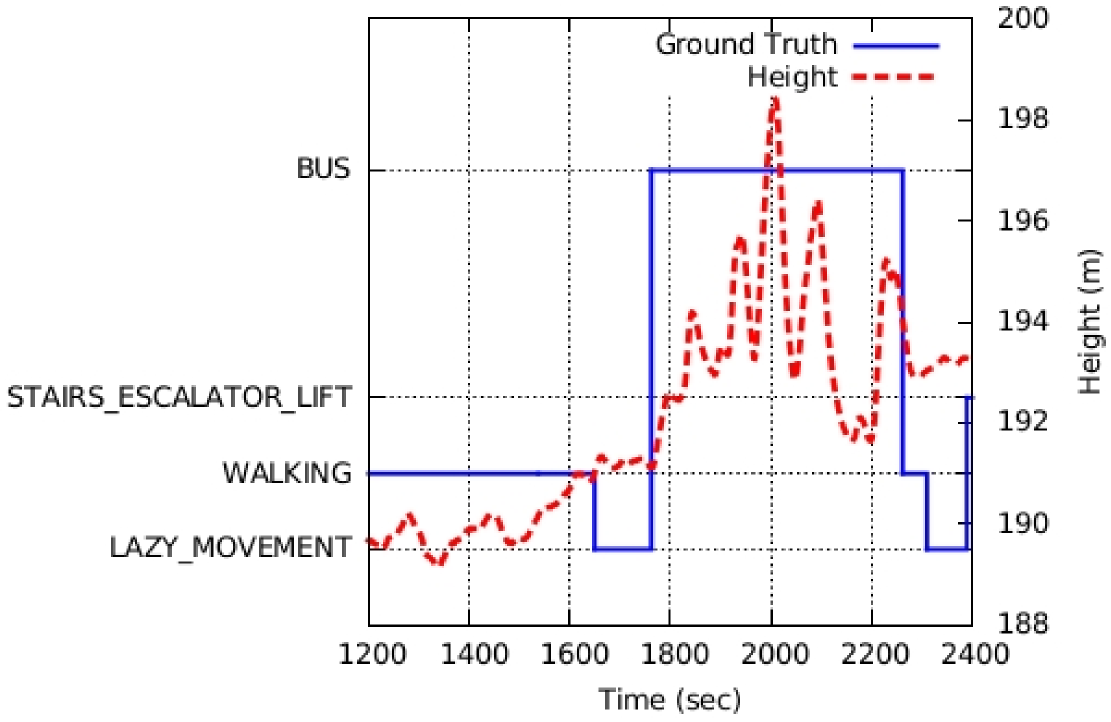
	Power (mW)	Increase over base power
Accl (20 Hz)	230	112%
Accl (10 Hz)	180	67%
Accl (2 Hz)	164	51%
Baro (1 Hz)	110	2%
CPU Awake (base)	108	x
CPU Asleep	25	x

The Grand Idea

Method Behind the Madness

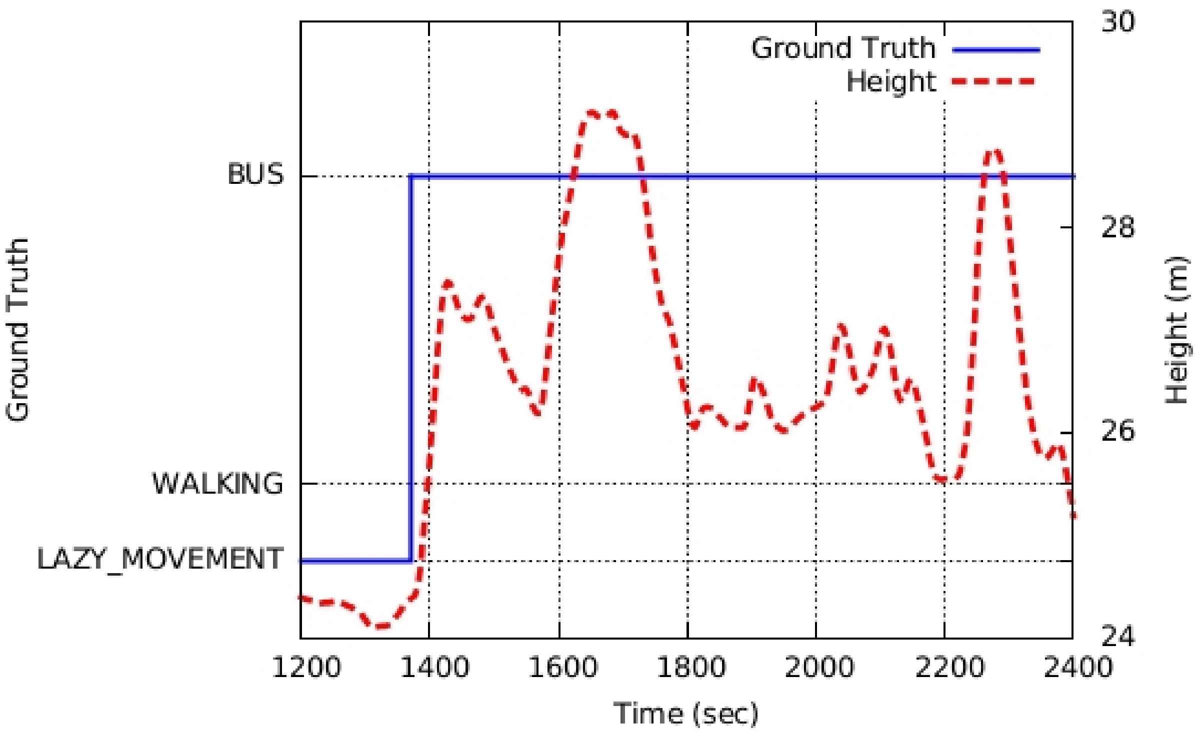
- 3 detected states: Idle, Walking, and Vehicle
  - Rate of height changes in a given time frame
  - States are less sensative than with accelerometer
  - Fewer false positives
- Can be used for tipping
  - Low granularity

A Picture is Worth…



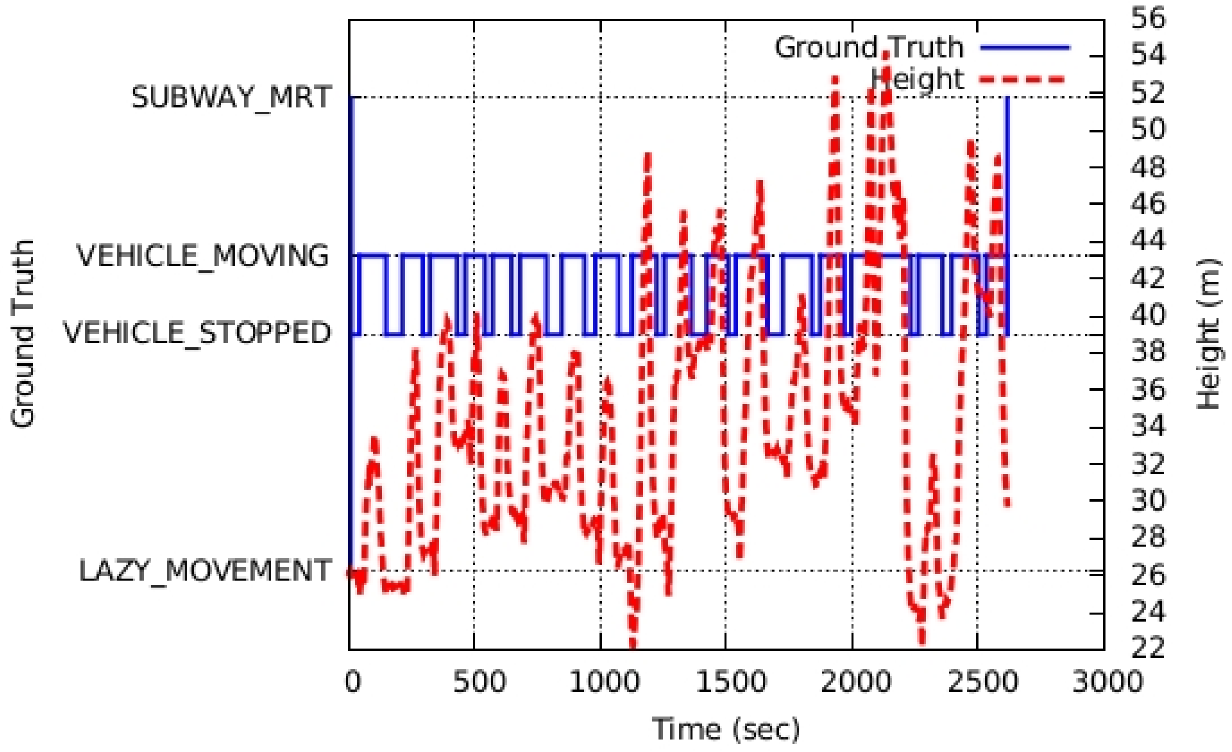
(a) Commute on a bus in Boston

A Picture is Worth…



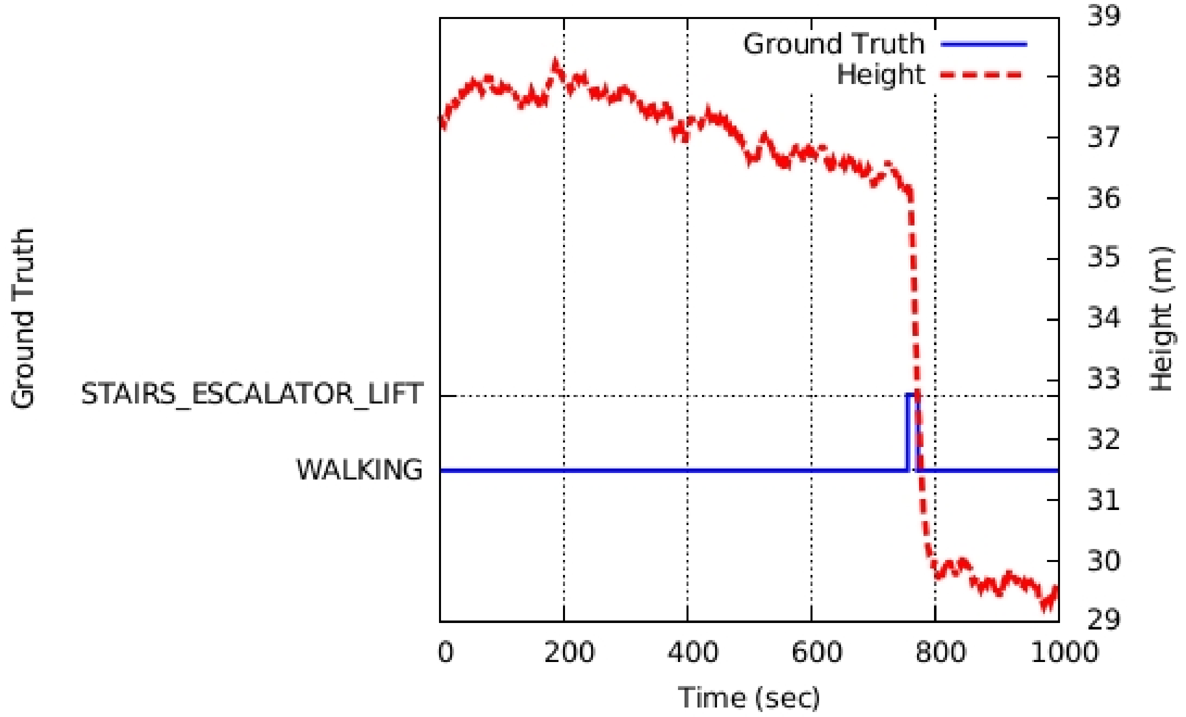
(b) Commute on a bus in Singapore

A Picture is Worth…



(c) Subway ride in Singapore

A Picture is Worth…



(d) Walking between floors in a mall

An Idea Made Real

System Overview

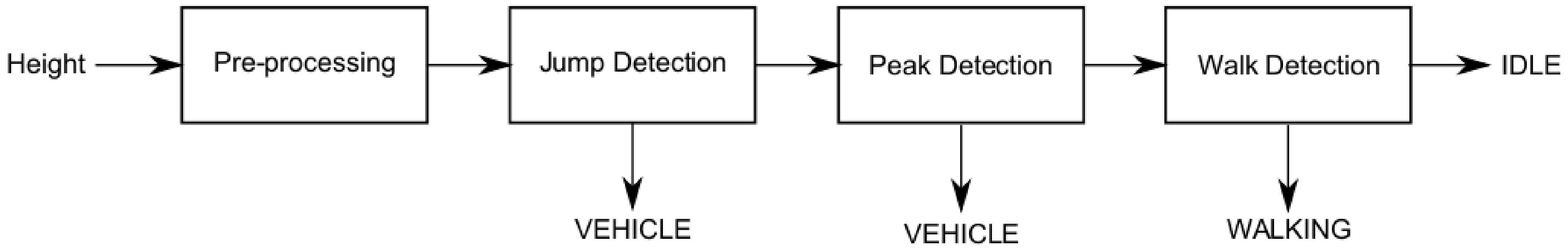


Figure 3: Overview of Barometer-based transportation context detection

System Overview

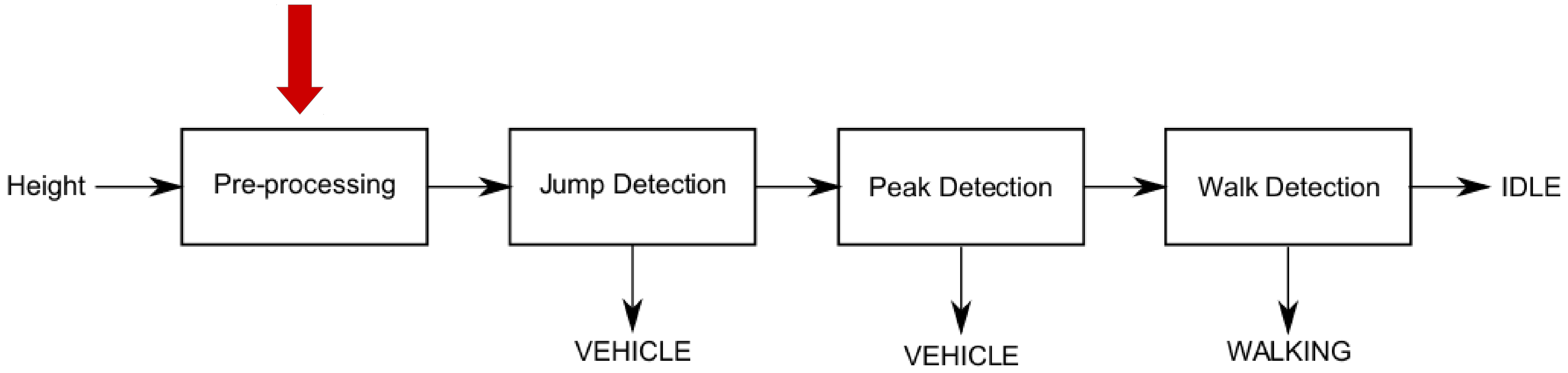


Figure 3: Overview of Barometer-based transportation context detection

Data Pre-processing

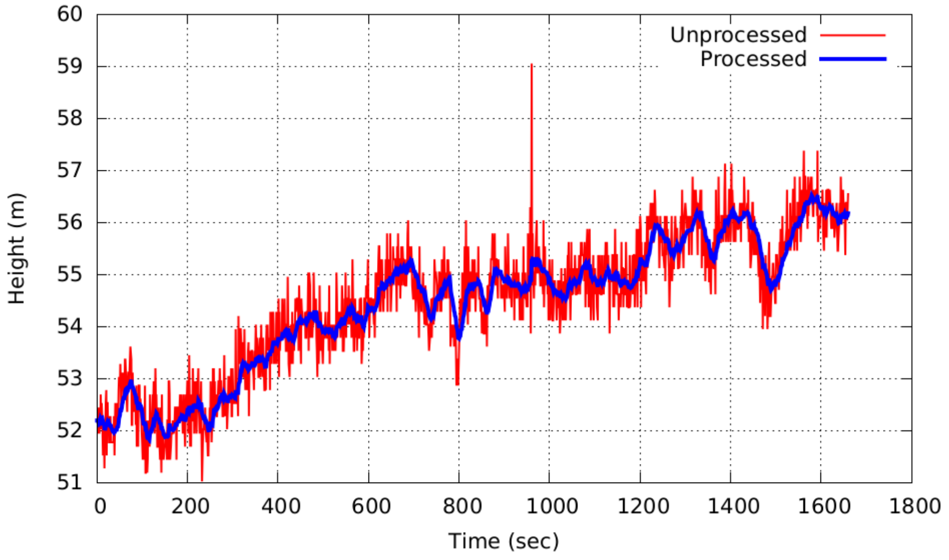


Figure 4: Barometer data after processing

- 1 Hz sampling rate
  - Software/hardware limit
- Data is noisy
  - Smoothing applied

$$currentHeight = \alpha * sensorHeight + (1 - \alpha) * prevHeight$$

JUMP DETection (Vehicle Detection)

- Jumps occur at high speeds or when traversing highly sloped roads.
  - > 0.8 m per 5 sec
- Track observed jumps in sliding 200 sec window.

- 30% - 70% ratio
  - Positive to negative
  - > 10 observations needed

PEAK DETection (Vehicle Detection)

- JumpDet fails for slow speeds on a non-highly sloped road.
  - Vehicles faster, experience more peaks
- Detects 1 m peaks & valleys
  - Simple, online algorithm
  - Run on every reading
- Track peaks & valleys in sliding 200 sec window
  - > 1 required
- Vehicle detections stitched together
  - < 2 min seperation
  - < 30 sec detections ignored

Walking and Idle Detections

- Default state: Idle
- Calculate standard deviation of height values in time window
  - Sliding 200 sec window
  - > 0.3 m is walking state
- Algorithm unaffected by weather drift.

Evaluation

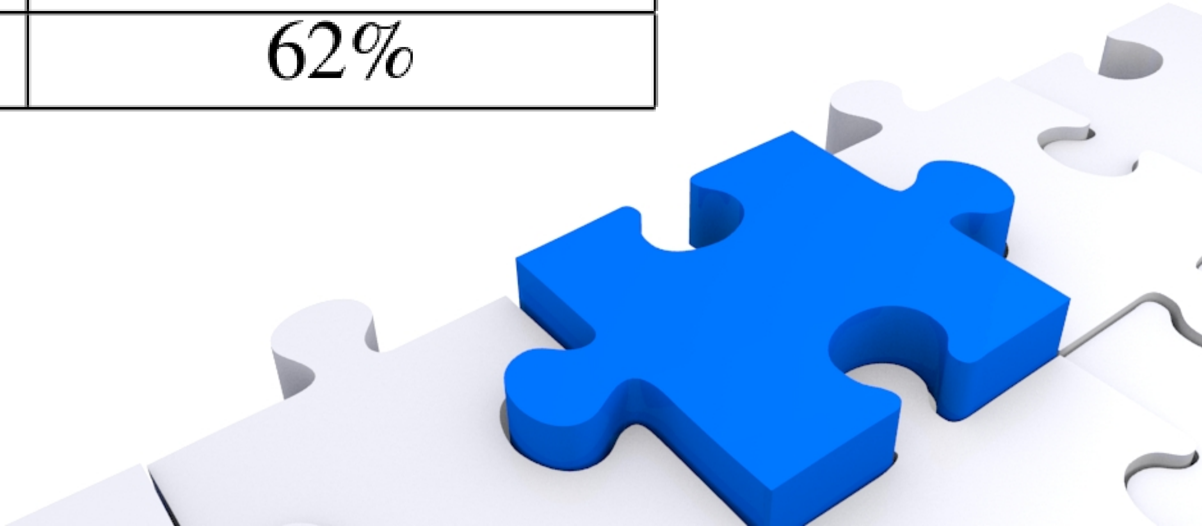
The Arena

Country	Volunteers	Total hours	Vehicle hours	Walking hours	Idle hours
Singapore	7	15	6.5	6.4	2.1
Boston (USA)	6	55.95	3.75	7.8	44.4
China	1	108.5	22	1.5	85

- Phones used: Nexus 4/5, Galaxy S3/S4
  - Android Jellybean (4.1 – 4.3)
- Special barometer traces for weather
- Barometer detection simulated

Accuracy

	Baro	FMS	Google	GoogleSmooth
Idle	76%	33%	76%	76%
Walking	54%	46%	79%	91%
Vehicle	81%	90%	31%	34%
Overall	69%	68%	56%	62%



Accuracy

	Baro	**Google**	**GoogleSmooth**
Idle	99%	97%	98%
Walking	23%	40%	50%
Vehicle	78%	24%	25%
Overall	93%	82%	83%

Accuracy by Location

	Singapore	**Boston**	**China**
Idle	76%	85%	99%
Walking	54%	40%	23%
Vehicle	81%	72%	78%
Overall	69%	79%	93%

Confusion Matrix

Table 7: Confusion Matrix for Barometer Algo

	Idle	Walking	Vehicle
Idle	76%	19%	5%
Walking	19%	54%	27%
Vehicle	6%	13%	81%

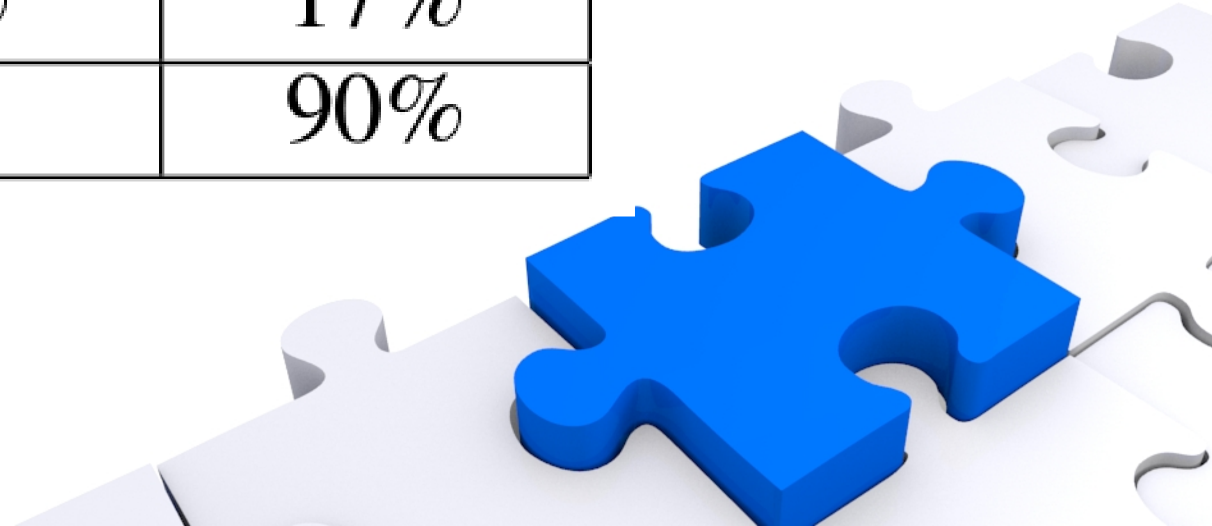
Table 8: Confusion Matrix for Google

	Idle	Walking	Vehicle	Unknown
Idle	76%	0%	0%	24%
Walking	10%	79%	0%	11%
Vehicle	38%	6%	31%	25%

Confusion Matrix

Table 9: Confusion Matrix for FMS

	Idle	Walking	Vehicle
Idle	33%	34%	33%
Walking	37%	46%	17%
Vehicle	6%	4%	90%



Latency

Table 13: Latency (sec) for each user state for barometer and Google algorithms (stddev in brackets)

	Baro	**Google**
Idle	176 (142)	78 (66)
Walking	158 (138)	26 (24)
Vehicle	211 (192)	122 (135)

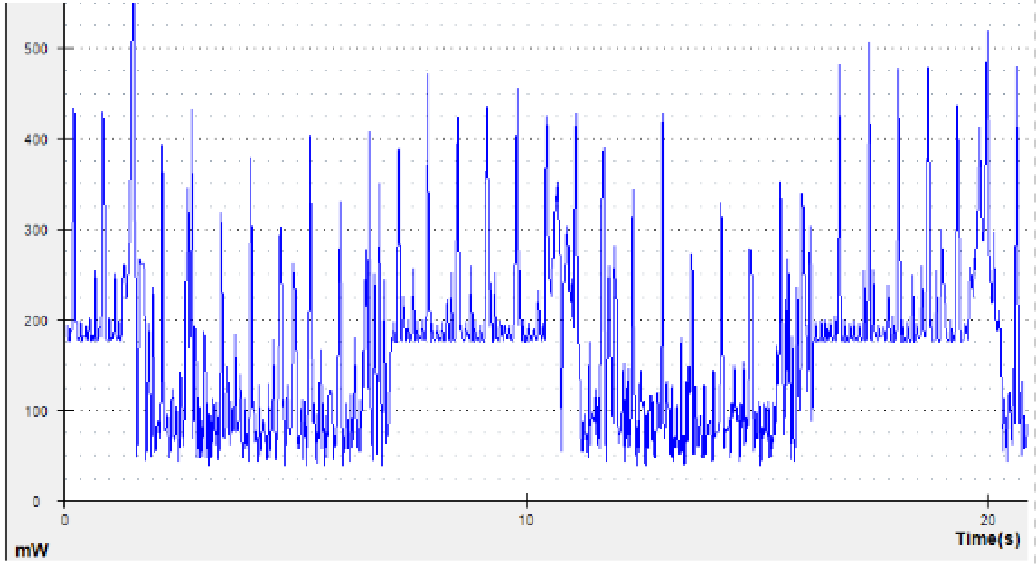
Power Usage

Table 14: Power usage

	Power (mW)
CPU Idle	25
CPU Awake	85
Google	120
Baro	88

Comparing the Power Levels

(a) Power profile of barometer-based context detection

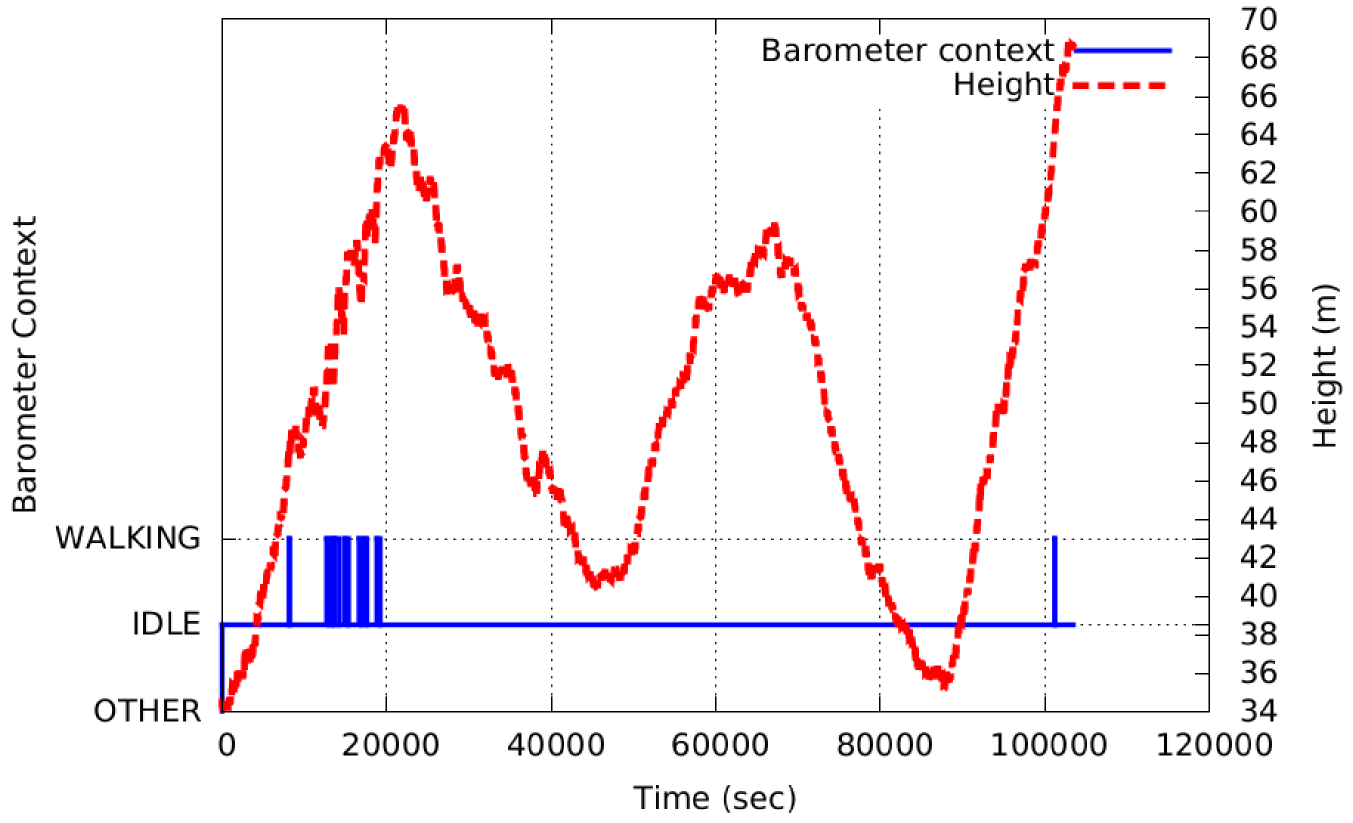


(b) Power profile of Google's context detection

Figure 10: Power profile of Google and barometer algorithms

Taking a Closer Look

Weathering Heights



Applicability Across Locations

Table 12: Comparison of terrain characteristics (stddev in brackets)

	Avg Elevation Change (m)	Avg Peak Distance (m)
Kansas City	0.84 (0.99)	479 (494)
San Francisco	1.05 (1.17)	645 (709)
Lausanne	1.04 (1.19)	395 (536)
Singapore	0.69 (0.65)	332 (252)
Boston	0.56 (0.66)	476 (435)

Applicability Across Locations

Table 11: Accuracy of barometer-based context detection algorithm using map elevation data at different speeds

	Vehicle (50 kmph)	Vehicle (35 kmph)	Vehicle (25 kmph)	Walk (5 kmph)	Walk (8 kmph)
Kansas City	96%	93%	89%	73%	56%
San Francisco	92%	90%	76%	74%	66%
Lausanne	84%	83%	79%	58%	50%
Singapore	99%	99%	98%	63%	32%
Boston	99%	97%	91%	66%	58%

Coming Together

A Meeting of the Minds

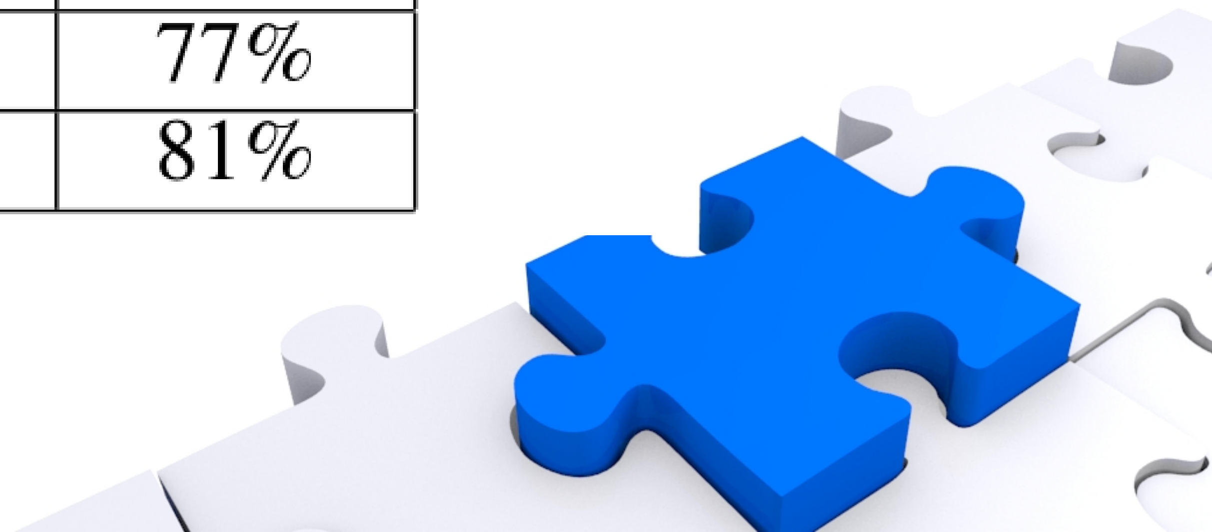
- Barometer
  - IDLE and certain VEHICLES
  - WALKING

- Google Activity Recognition
  - WALKING
  - IDLE and certain VEHICLES

Two Minds Are Better Than One

Table 15: Fusing barometer and Google algorithms

	Baro	**Google**	**Fusion**
Idle	76%	76%	76%
Walking	54%	79%	88%
Vehicle	81%	31%	77%
Overall	69%	56%	81%



Conclusion

- Allows for activity detection with lower power usage compared to accelerometer.
- Classifies three kinds of movement states: IDLE, WALKING, and VEHICLE.
- A good method for detecting IDLE states and fast vehicle movement.
  - Poor WALKING detection.

Discussion

- What are some other ways to correct the issue with the barometer's WALKING detection?
- Are the three categories of IDLE, WALKING, and VEHICLE enough?
- Does this approach have any real value?

Android application – Cover
- needs context to make is possible.
Context for location readings.

Hardware:
M7/M8 Co-processor
- Collects and process from accelerometer, gyroscope, and compasses
- IOS
Step counter
- Android
- Coprocessor

Transportation-mode Detection

- Useful for many things
  - Personal tracking
  - Regional data/planning
- Usually accelerometers
  - Low power
  - 10+ Hz sampling
  - Complex ML model

Android Move
- First to use Google's Activity Recognition API.

A Flash

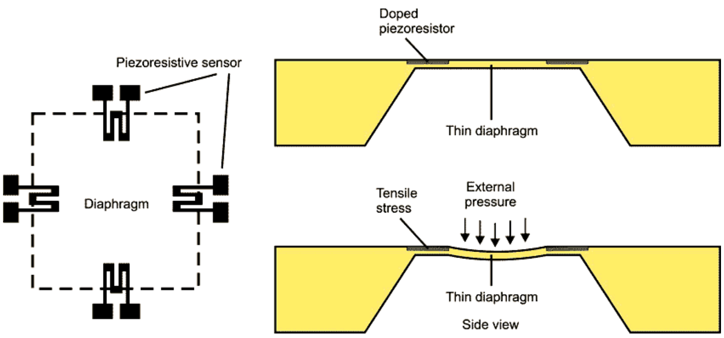
- Terrain is not perfectly flat
- Faster transportation = faster traversal of terrain
- More vertical movement for faster modes of transporation

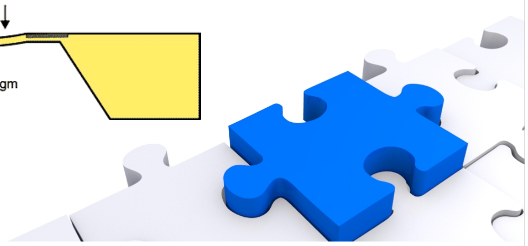
Barometer Background

- Orientation and position independent
- Simpler calibration
- Better WAIT detection
- Lower necessary sampling rate

- Barometric pressure unstable
  - Altitude
  - Weather
  - Temperature
- Installation bias
- Aging drift

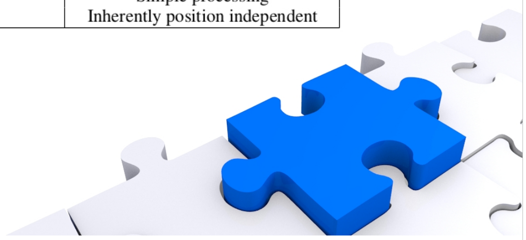
Barometer Breakdown





Methodology Comparisons

Sensor	**Power**	**Limitations**	**Barometer advantage**
GPS	Very high	Lack of indoor/underground coverage High power usage	Usable everywhere Ultra-low-power
WiFi/Cellular	High/Moderate	Requires dense access points/cellular towers	No external infrastructure
Accelerometer	Low	Extensive training required Classification complexity Position dependence	Simple calibration based on terrain Simple processing Inherently position independent



Power Play

	Power (mW)	Increase over base power
Accl (20 Hz)	230	112%
Accl (10 Hz)	180	67%
Accl (2 Hz)	164	51%
Baro (1 Hz)	110	2%
CPU Awake (base)	108	x
CPU Asleep	25	x

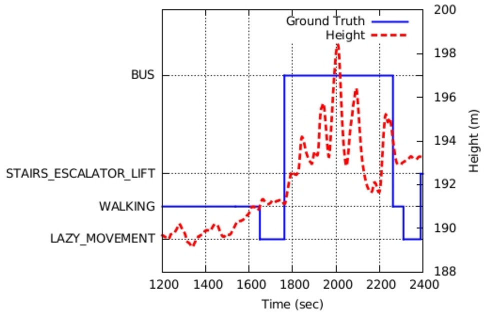
Power Play

	Power (mW)	Increase over base power
Accl (20 Hz)	230	112%
Accl (10 Hz)	180	67%
Accl (2 Hz)	164	51%
Baro (1 Hz)	110	2%
CPU Awake (base)	108	x
CPU Asleep	25	x

Method Behind the Madness

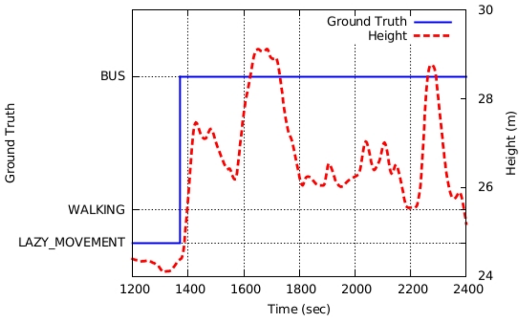
- 3 detected states: Idle, Walking, and Vehicle
  - Rate of height changes in a given time frame
  - States are less sensative than with accelerometer
  - Fewer false positives
- Can be used for tipping
  - Low granularity

A Picture is Worth...



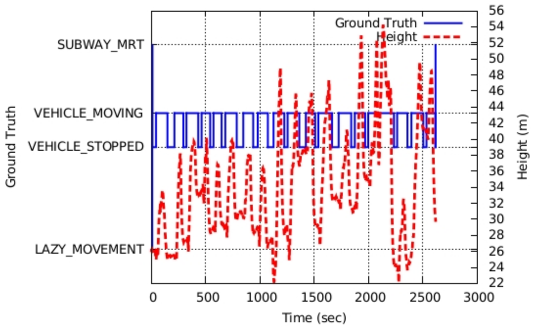
(a) Commute on a bus in Boston

A Picture is Worth…



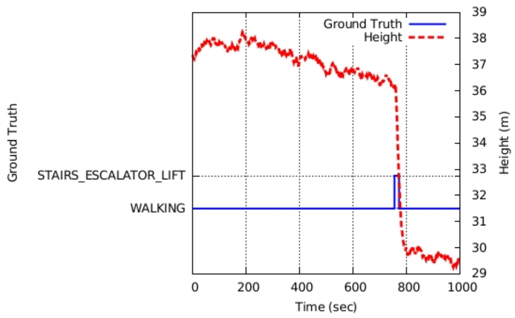
(b) Commute on a bus in Singapore

A Picture is Worth…



(c) Subway ride in Singapore

A Picture is Worth...



(d) Walking between floors in a mall

System Overview

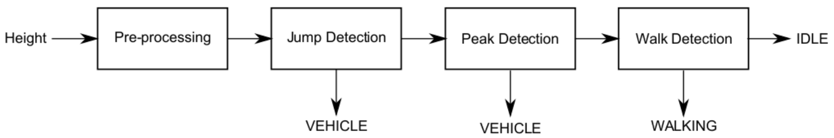


Figure 3: Overview of Barometer-based transportation context detection

System Overview

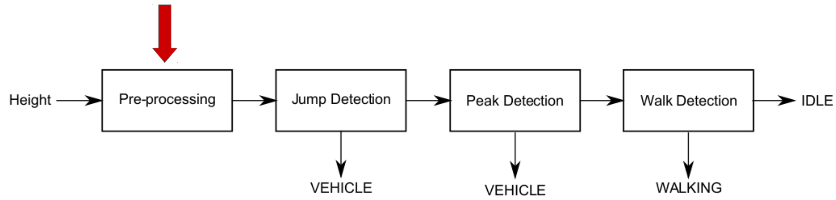


Figure 3: Overview of Barometer-based transportation context detection

Data Pre-processing

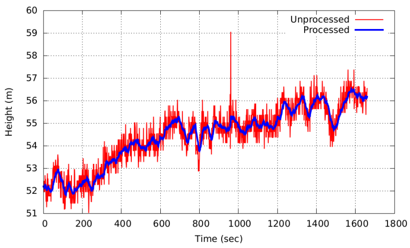


Figure 4: Barometer data after processing

- 1 Hz sampling rate
  - Software/hardware limit
- Data is noisy
  - Smoothing applied

$$currentHeight = \alpha * sensorHeight + (1 - \alpha) * prevHeight$$

JUMP DETection (Vehicle Detection)

- Jumps occur at high speeds or when traversing highly sloped roads.
  - > 0.8 m per 5 sec
- Track observed jumps in sliding 200 sec window.

- 30% - 70% ratio
  - Positive to negative
  - > 10 observations needed

PEAK DETection (Vehicle Detection)

- JumpDet fails for slow speeds on a non-highly sloped road.
  - Vehicles faster, experience more peaks
- Detects 1 m peaks & valleys
  - Simple, online algorithm
  - Run on every reading
- Track peaks & valleys in sliding 200 sec window
  - > 1 required
- Vehicle detections stitched together
  - < 2 min seperation
  - < 30 sec detections ignored

Walking and Idle Detections

- Default state: Idle
- Calculate standard deviation of height values in time window
  - Sliding 200 sec window
  - > 0.3 m is walking state
- Algorithm unaffected by weather drift.

The Arena

Country	Volunteers	Total hours	Vehicle hours	Walking hours	Idle hours
Singapore	7	15	6.5	6.4	2.1
Boston (USA)	6	55.95	3.75	7.8	44.4
China	1	108.5	22	1.5	85

- Phones used: Nexus 4/5, Galaxy S3/S4
  - Android Jellybean (4.1 – 4.3)
- Special barometer traces for weather
- Barometer detection simulated

178 hours of traces (47 transportation)
Galaxy x3 samples at 5 Hz
Nexus 4 samples at 4 Hz
Nexus 5 (internal smoothing) value every 2 sec

Accuracy

	Baro	FMS	Google	GoogleSmooth
Idle	76%	33%	76%	76%
Walking	54%	46%	79%	91%
Vehicle	81%	90%	31%	34%
Overall	69%	68%	56%	62%

In Singapore

Accuracy

	Baro	**Google**	**GoogleSmooth**
Idle	99%	97%	98%
Walking	23%	40%	50%
Vehicle	78%	24%	25%
Overall	93%	82%	83%

In China

Accuracy by Location

	Singapore	Boston	China
Idle	76%	85%	99%
Walking	54%	40%	23%
Vehicle	81%	72%	78%
Overall	69%	79%	93%

China
– cross country train ride
- walking ground truth possibly inaccurate

Boston and Singapore
- inaccurate traces disregarded

Confusion Matrix

Table 7: Confusion Matrix for Barometer Algo

	Idle	**Walking**	**Vehicle**
Idle	76%	19%	5%
Walking	19%	54%	27%
Vehicle	6%	13%	81%

Table 8: Confusion Matrix for Google

	Idle	**Walking**	**Vehicle**	**Unknown**
Idle	76%	0%	0%	24%
Walking	10%	79%	0%	11%
Vehicle	38%	6%	31%	25%

Confusion Matrix

Table 9: Confusion Matrix for FMS

	Idle	**Walking**	**Vehicle**
Idle	33%	34%	33%
Walking	37%	46%	17%
Vehicle	6%	4%	90%

Latency

Table 13: Latency (sec) for each user state for barometer and Google algorithms (stddev in brackets)

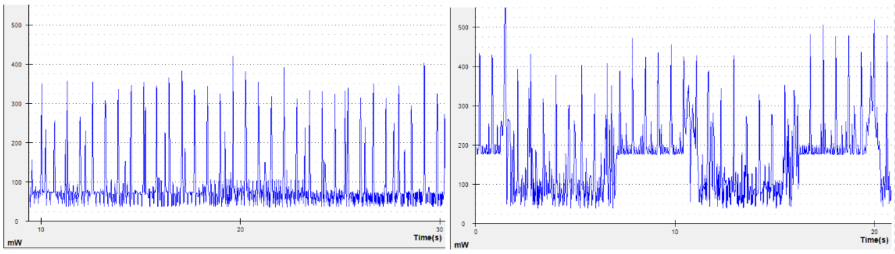
	Baro	**Google**
Idle	176 (142)	78 (66)
Walking	158 (138)	26 (24)
Vehicle	211 (192)	122 (135)

Power Usage

Table 14: Power usage

	Power (mW)
CPU Idle	25
CPU Awake	85
Google	120
Baro	88

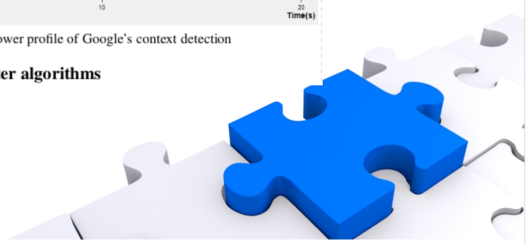
Comparing the Power Levels

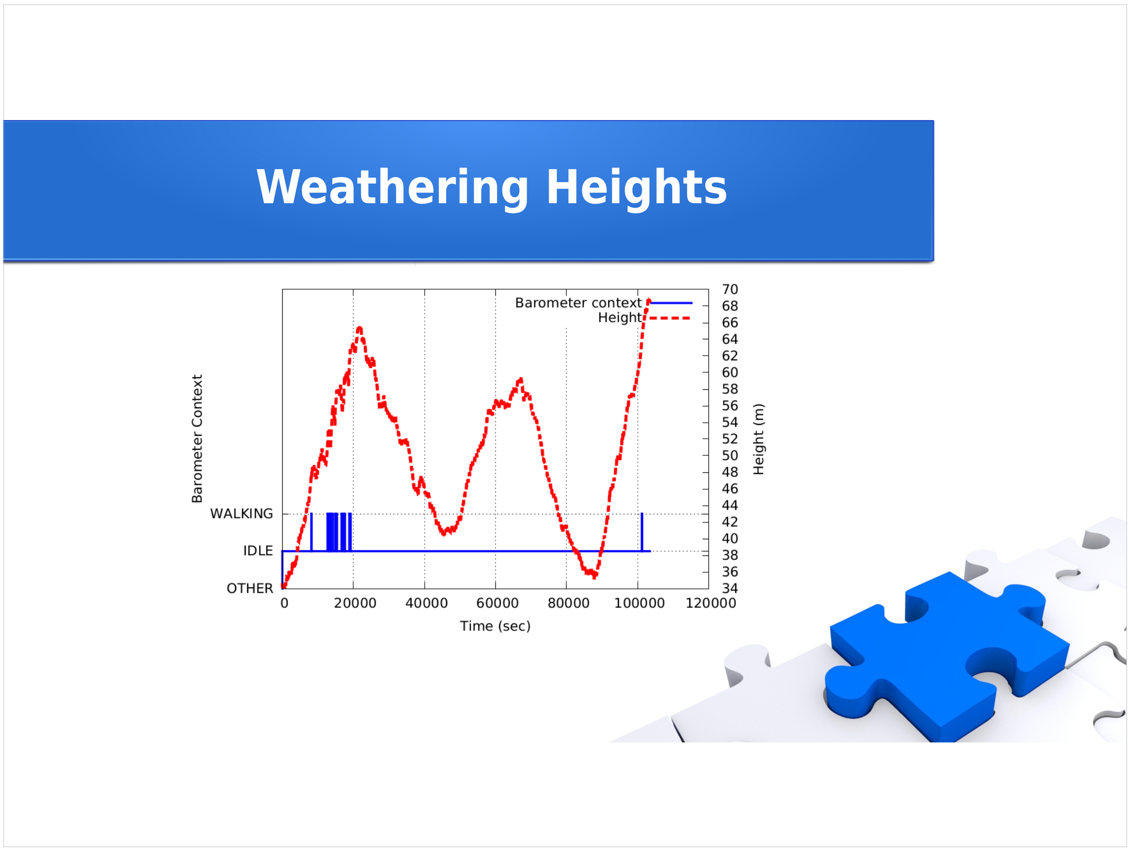


(a) Power profile of barometer-based context detection

(b) Power profile of Google's context detection

Figure 10: Power profile of Google and barometer algorithms





Weather variance during eval traces in Boston and Singapore

Tested only against IDLE

Collected in Singapore for rainy and windy days

Applicability Across Locations

Table 12: Comparison of terrain characteristics (stddev in brackets)

	Avg Elevation Change (m)	**Avg Peak Distance (m)**
Kansas City	0.84 (0.99)	479 (494)
San Francisco	1.05 (1.17)	645 (709)
Lausanne	1.04 (1.19)	395 (536)
Singapore	0.69 (0.65)	332 (252)
Boston	0.56 (0.66)	476 (435)

Applicability Across Locations

Table 11: Accuracy of barometer-based context detection algorithm using map elevation data at different speeds

	Vehicle (50 kmph)	**Vehicle (35 kmph)**	**Vehicle (25 kmph)**	**Walk (5 kmph)**	**Walk (8 kmph)**
Kansas City	96%	93%	89%	73%	56%
San Francisco	92%	90%	76%	74%	66%
Lausanne	84%	83%	79%	58%	50%
Singapore	99%	99%	98%	63%	32%
Boston	99%	97%	91%	66%	58%

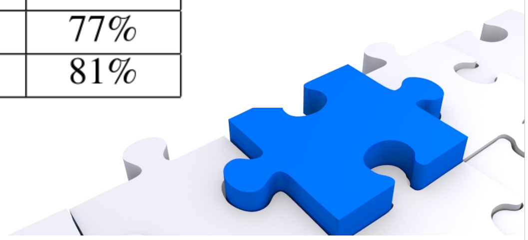
A Meeting of the Minds

- Barometer
  - IDLE and certain VEHICLES
  - WALKING
- Google Activity Recognition
  - WALKING
  - IDLE and certain VEHICLES

Two Minds Are Better Than One

Table 15: Fusing barometer and Google algorithms

	Baro	**Google**	**Fusion**
Idle	76%	76%	76%
Walking	54%	79%	88%
Vehicle	81%	31%	77%
Overall	69%	56%	81%



Conclusion

- Allows for activity detection with lower power usage compared to accelerometer.
- Classifies three kinds of movement states: IDLE, WALKING, and VEHICLE.
- A good method for detecting IDLE states and fast vehicle movement.
  - Poor WALKING detection.

Discussion

- What are some other ways to correct the issue with the barometer's WALKING detection?
- Are the three categories of IDLE, WALKING, and VEHICLE enough?
- Does this approach have any real value?