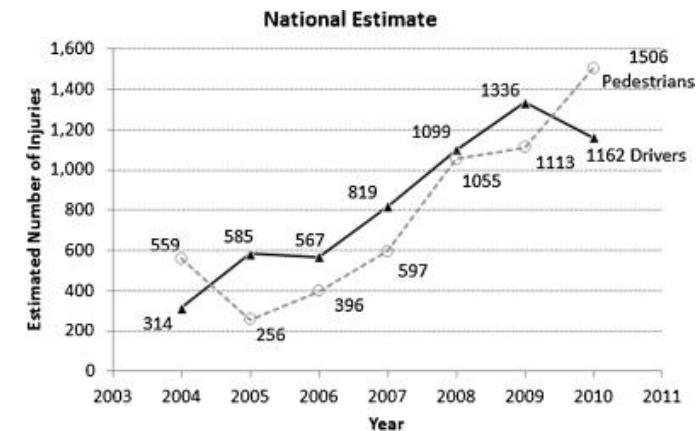
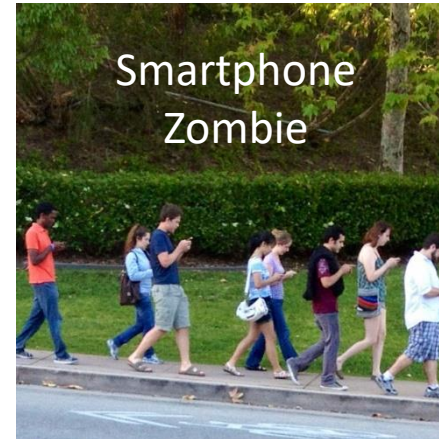


Auto++: Detecting Cars Using Embedded Microphones in Real-Time

Sugang Li, Xiaoran Fan, Yanyong Zhang, Wade Trappe, Janne Lindqvist,
Richard E. Howard

Motivation

- ❑ “Smombie” was chosen as the official “Youth Word” in Germany in 2015
- ❑ In the U.S., A pedestrian was killed every two hours, and injured every 8 minutes[1].
- ❑ From 2004 to 2010, the phone related injuries increased from 559 to 1506[2].



[1] NHTSA. Trac safety facts 2013. DOT HS 812 139.

[2] Nasar, Jack L., and Derek Troyer. "Pedestrian injuries due to mobile phone use in public places." Accident Analysis & Prevention 57 (2013): 91-95.

Existing Solutions

☐ Infrastructure Support



☐ Mobile Technology Support



DSRC, LIDAR & Camera



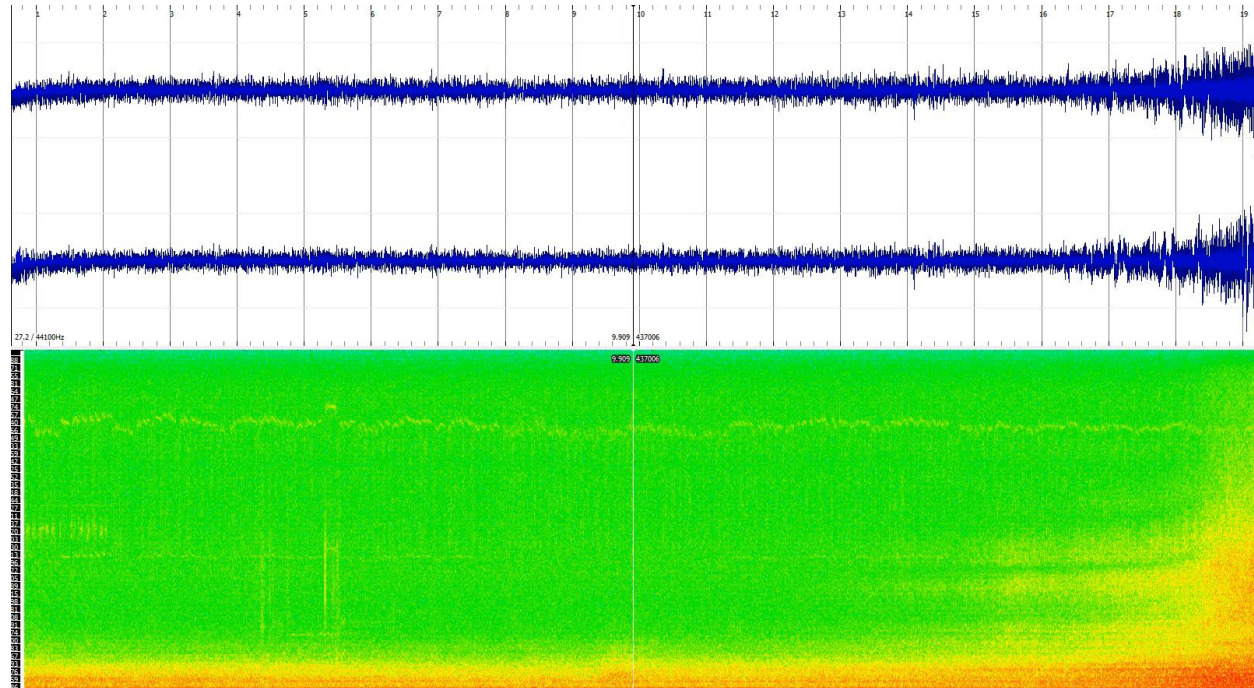
Smartphone CV (Walksafe 12')

Potential Applications

- Pedestrian Safety
- Residential Area Traffic monitoring
- Enriching Augmented Reality Game

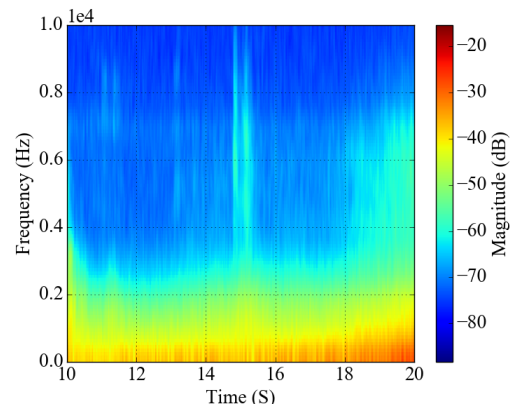


Challenges

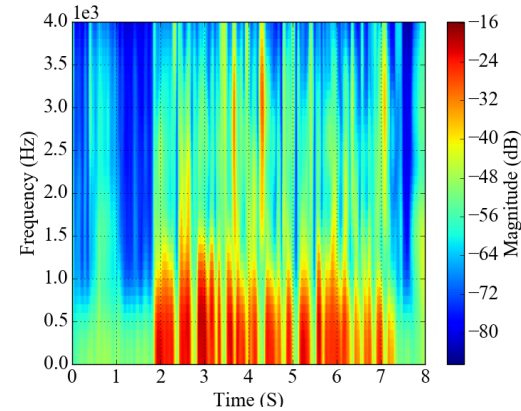


- No instantaneous cue for car's presence
- No outstanding acoustic feature

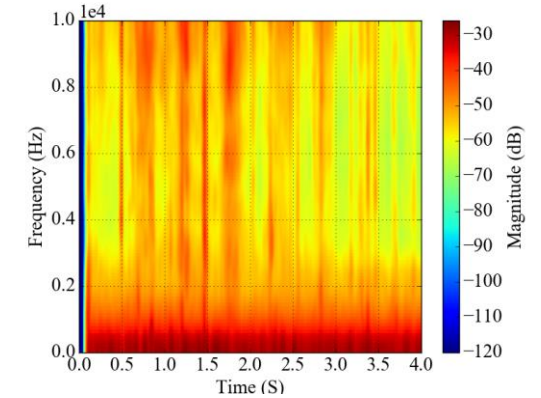
Challenges



Approaching Car



Speech



Steps

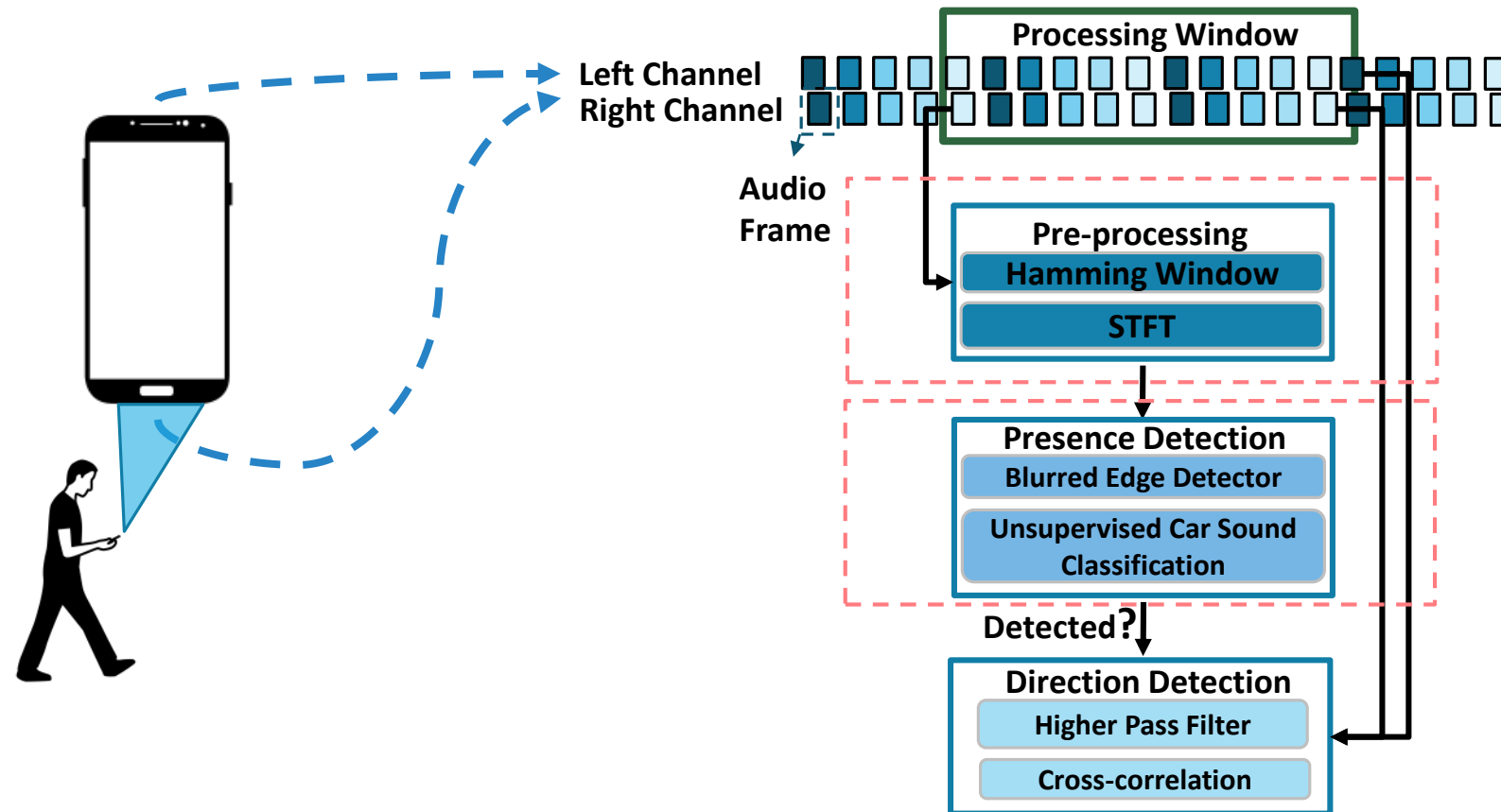
The nature of car sound

- Tire-pavement friction noise
- No obvious acoustic formant

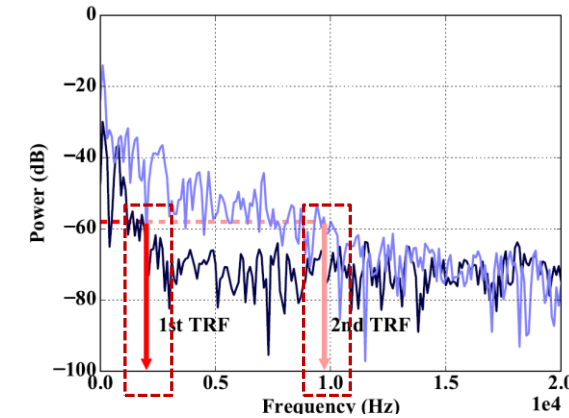
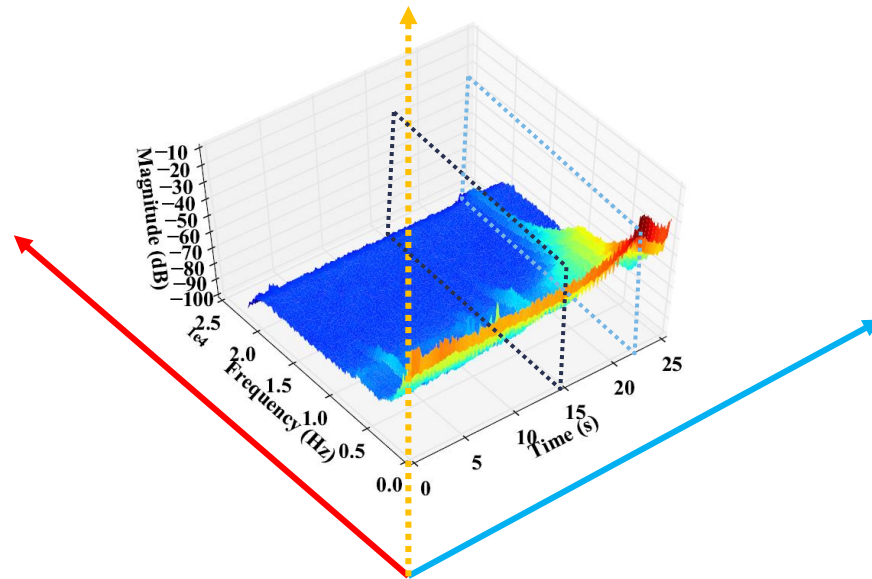
Existing acoustic techniques are not feasible

- Doppler Shift
- MFCC

Auto++ Design



Presence Detection : Top-right Frequency (TRF)



Free space path-loss:

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2$$

Top-Right Frequency (TRF): Maximum Frequency Whose Power Reaches a Certain Threshold

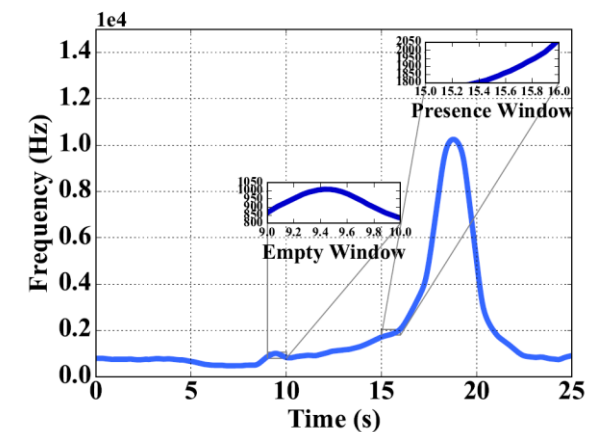
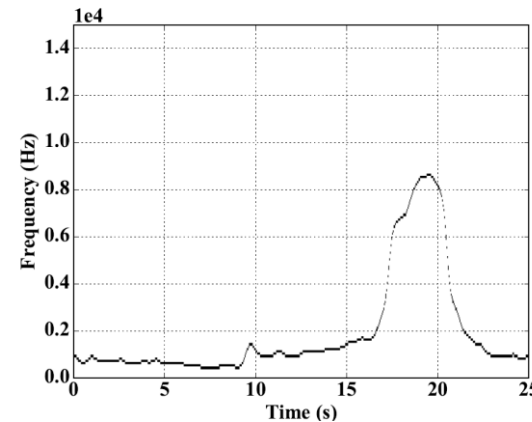
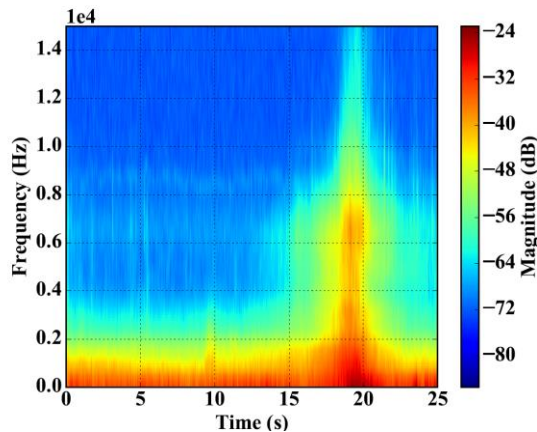
$$TRF(n) = \max\{\operatorname{argmax}_f(S(n, f))\}$$

Presence Detection: Feature Extraction

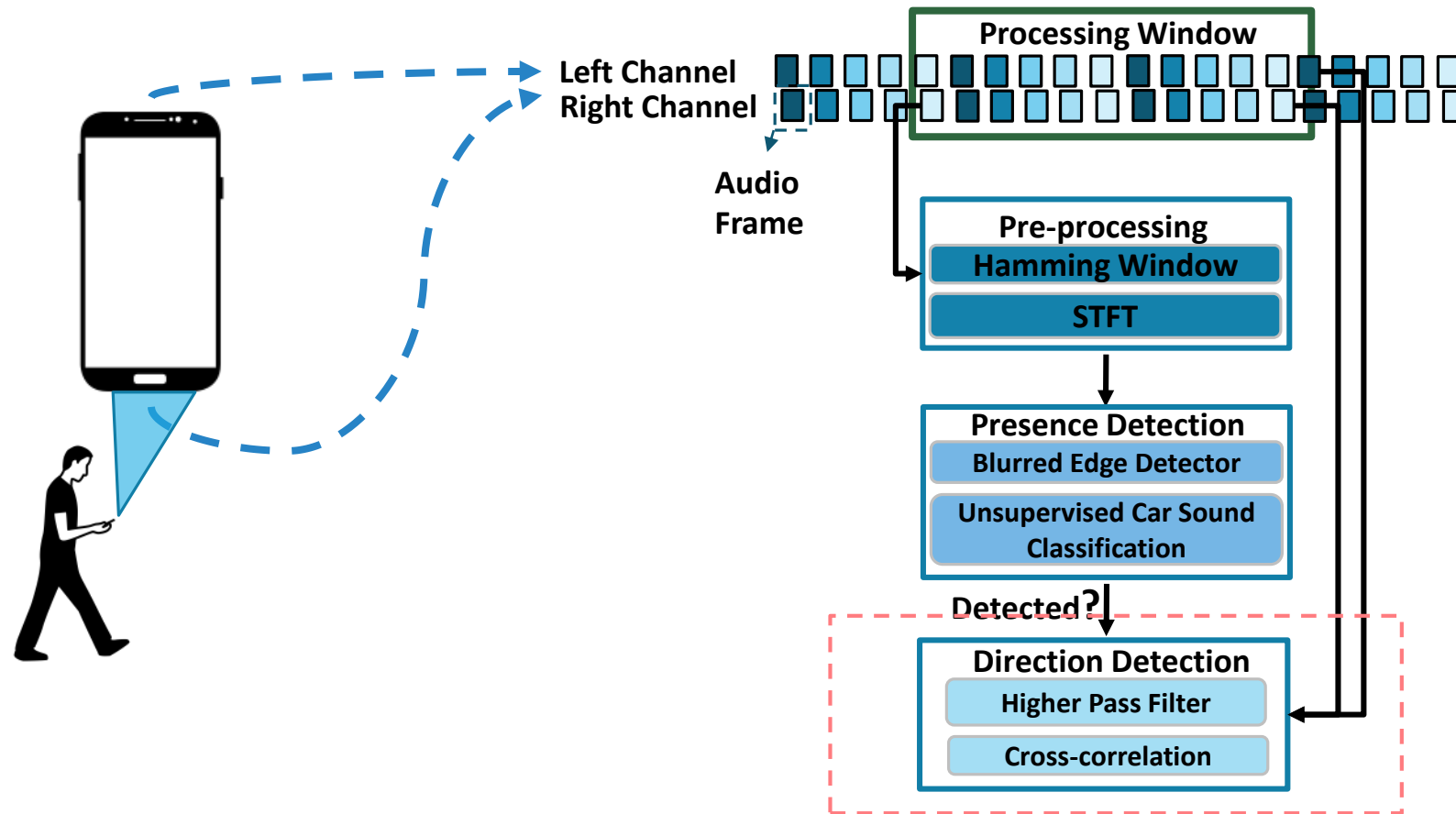
Edge Detection



Edge Detection for TRF



Auto++ Design



Direction Estimation

□ TDoA estimation

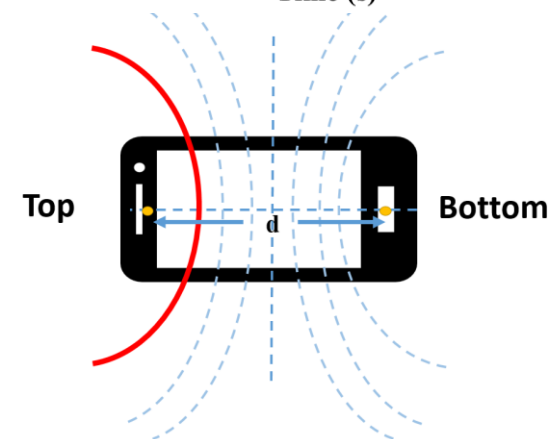
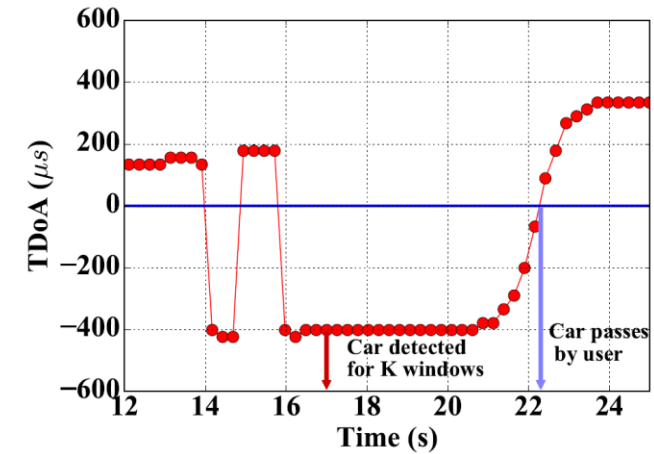
- Cross-correlation function of 2 channel input

$$(f * g)[\tau] = \sum_{t=-\infty}^{\infty} f^*[t]g[t + \tau]$$

- Find the lag with the maximum value

□ SPL is sensitive to noise

- Trigger only if we detect a presence event



Evaluation: Overview

□ Objectives

- Presence detection's timeliness & accuracy
- Direction detection's accuracy

□ Wide range of test cars

- 7 models – sedan, SUV, electric car, sport car

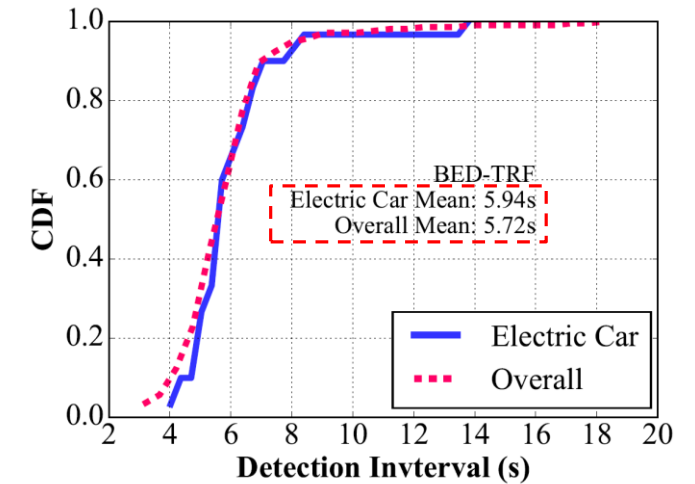
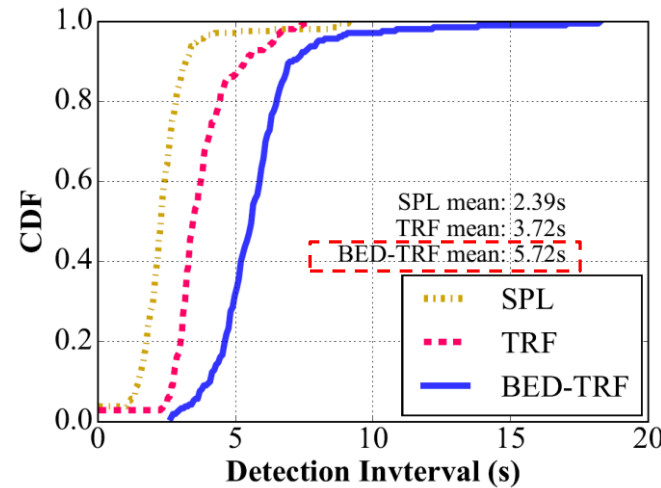
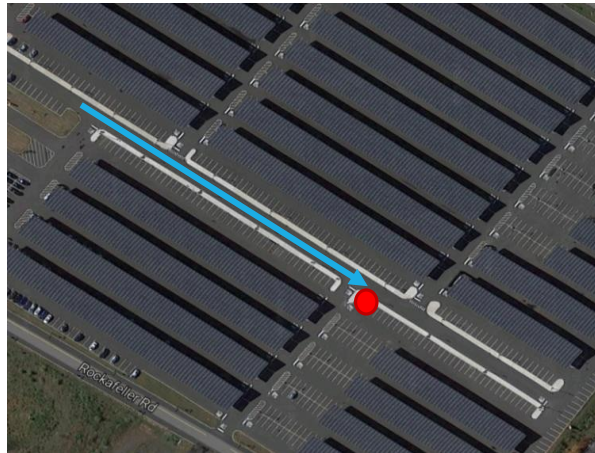
□ Various environments

- Parking lot, residential area, campus road, shopping center



Evaluation: Presence Detection

Timeliness



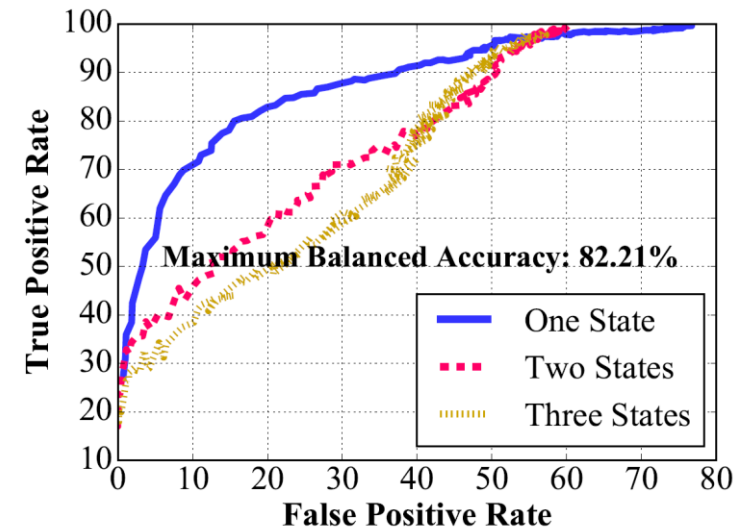
- ✓ Edge detection based algorithm outperforms the SPL baseline algorithm and the naïve TRF algorithm
- ✓ Electric car can be detected

Evaluation: Presence Detection

Accuracy

	Quiet Parking Lot	Campus Road	Residential Area	Shopping Center
Detection Interval (s)	5.7	4.2	3.6	3.2
TPR(%)	100	97.2	94.8	83.8
FPR(%)	0	3.3	13.3	13.3

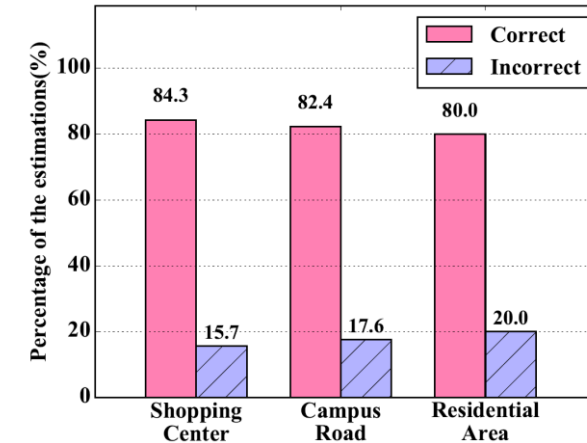
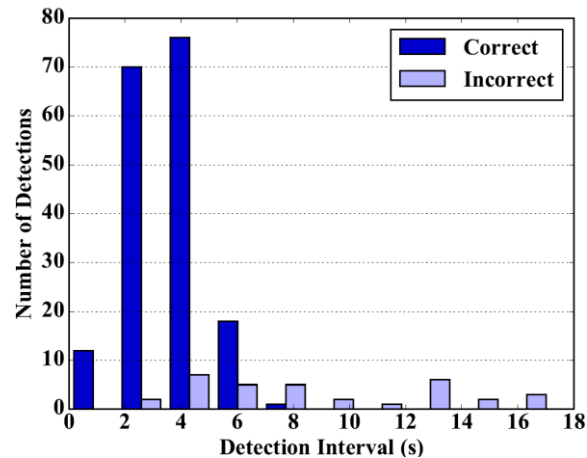
- ✓ Auto++ is highly accurate in less crowded area
- ✓ Auto++ can still detect cars in challenged environment
- ✓ Auto++ outperforms MFCC based solution



MFCC-based Results

Evaluation: Direction Detection

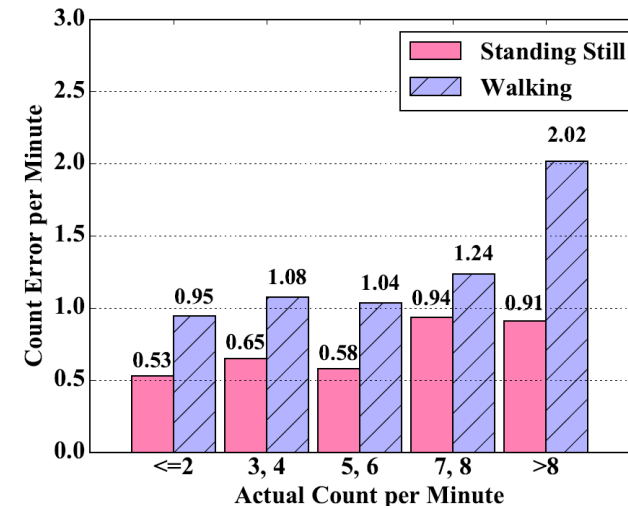
Accuracy



- ✓ The accuracy of direction detection is increasing as the car is getting closed
- ✓ It is robust in various noisy environment

Evaluation: Car Count

□ Error count



- ✓ Auto++ has low count error in suburban residential area
- ✓ It is robust to user's activity

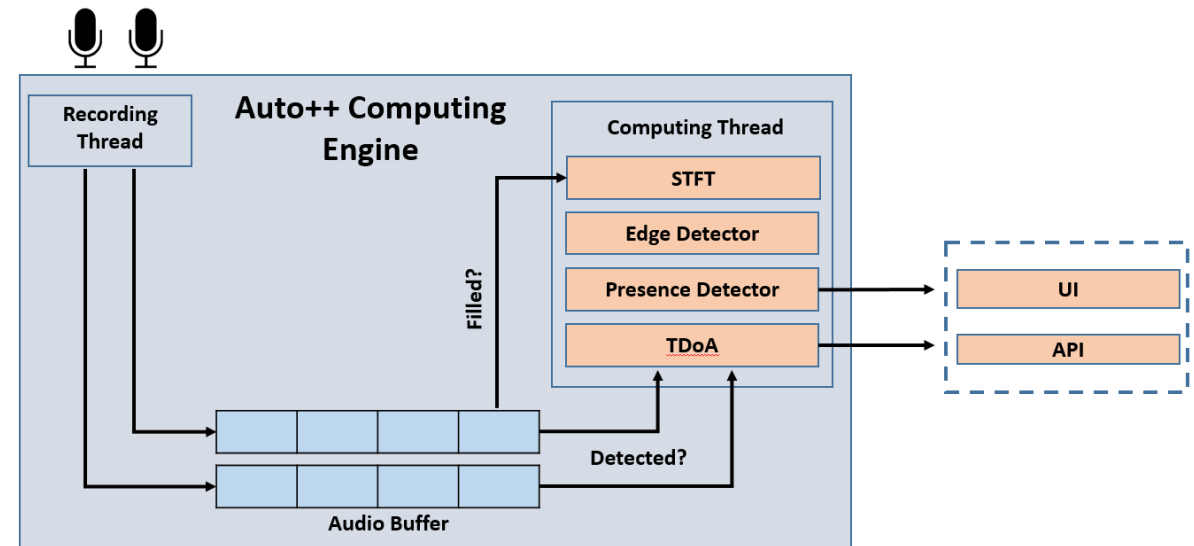
Prototype

Platform

- Android + OpenCV
- Concurrent Threads

Memory and CPU usage profiling

- Memory does not exceed the OS pre-allocated size
- Peak CPU usage is only 1.8%



	Nexus 6/6P (Idling)	Nexus 6/6P (Presence Detection)	Nexus 6/6P (Direction Estimation)
Memory (MB)	36.4/32.8	36.4/32.8	36.4/32.8
CPU (%)	0/0	2/1.1	3.2/1.8

Conclusion

- ❑ We proposed Auto++, a vehicle sensing system that used off-the-self smartphone microphones
- ❑ We introduced a new feature (TRF) that caught the presence of a vehicle timely and accurately
- ❑ We proposed a edge detection based algorithm to extract traces robustly
- ❑ We conducted comprehensive experiments to evaluate the performance of Auto++'s presence detection, direction detection, and car counting.
- ❑ We implemented Auto++ on Android platform and showed its feasibility as a background service.

Thank you!

 Q & A