Turning a Mobile Device into a Mouse in the Air

Sangki Yun, **Yi-Chao Chen and Lili Qiu**

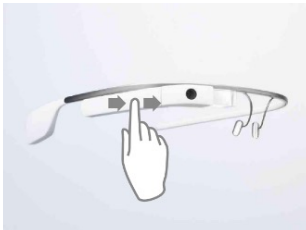
Department of Computer Science

The University of Texas at Austin

ACM MobiSys 2015, Florence, Italy

Era of Smart Devices

Smart Devices



Ideally

Our goal

- Controlling smart devices using **existing** mobile devices

How to enable precise mobile device position tracking?

Possible solutions

- RF signal based device localization – sub-meter level accuracy
- Accelerometer based device tracking – not feasible due to huge noise
- Gyroscope – track the rotational movement of the device

Our Approach

- Audio based movement tracking using Doppler effect

- Two speakers emit pure sinusoid tones in inaudible frequencies

- The device position is tracked by the Doppler shift of the recorded audio signal

Outline

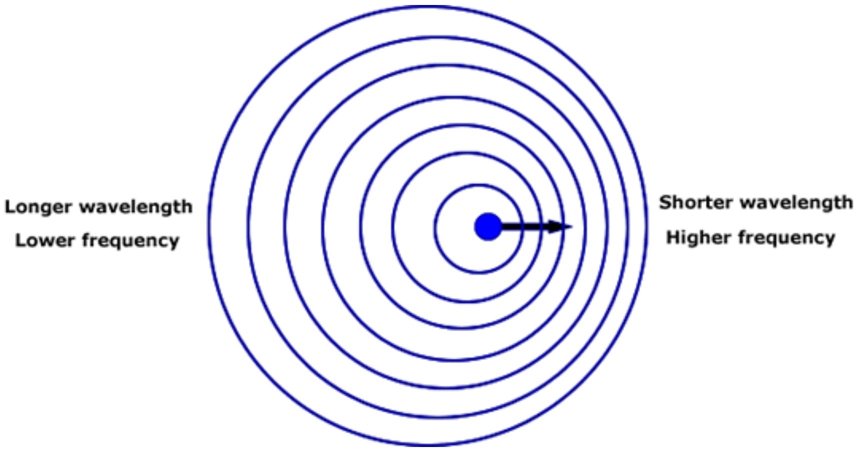
- Background

Doppler Effect

- Doppler shift – change in the frequency due to the movement of the sender or the receiver

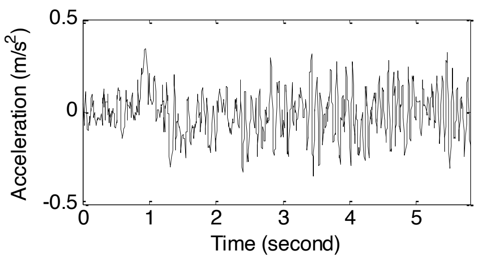
$$F^S = \frac{v}{c}F \quad \text{-->} \quad v = \frac{F^S}{F}c$$

F^S: Doppler shift
c: propagation speed of the medium
F : frequency of the wave

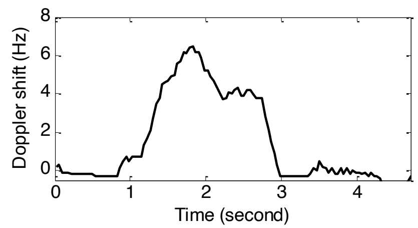


Why Doppler? (Why not accelerometer?)

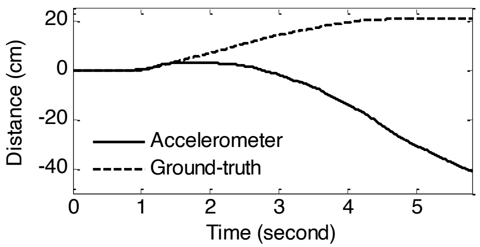
- Compared to accelerometer, it is much more accurate to track the movement



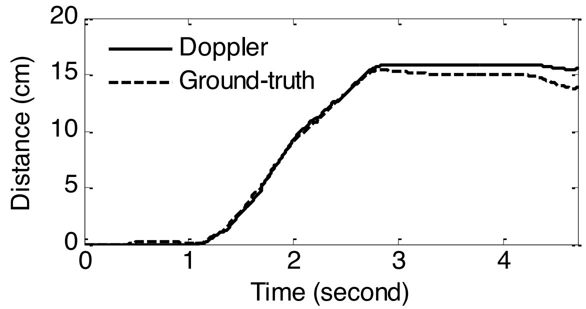
Acceleration while device is moving



Measured Doppler shift



Movement estimated from the accelerometer



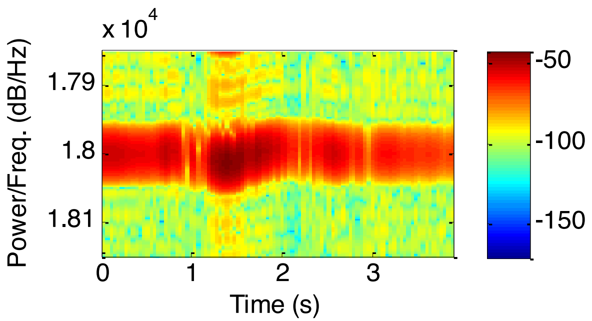
Movement estimated from Doppler shift

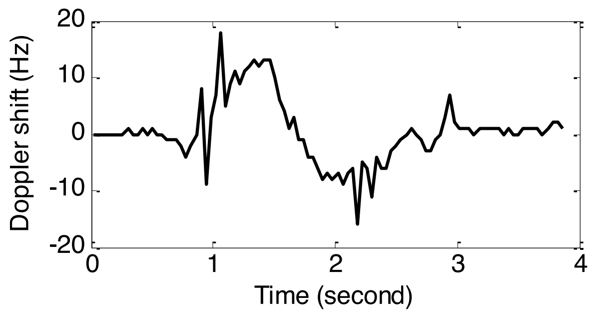
Doppler based device movement tracking

- Accurately estimating the Doppler shift
- Tracking the position from the distance change
- Finding the distance between speakers
- Finding the initial position of the device
- Controlling a device with one speaker

Doppler shift estimation

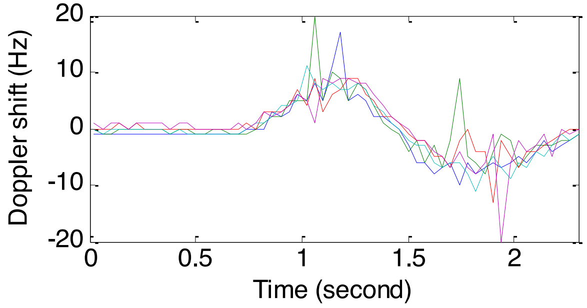
- Short-term Fourier Transform (STFT)
  - Observe the change of spectrum over time
  - Use Hamming window to mitigate aliasing
- Detect the Doppler shift by finding the peak frequency with maximum magnitude



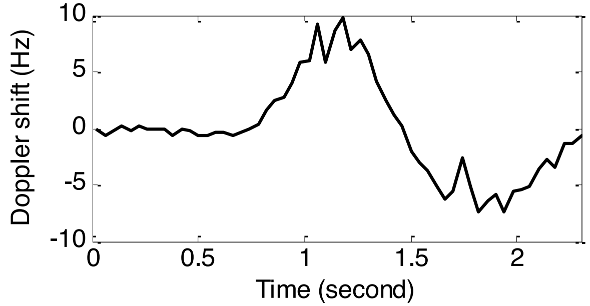


Improving the frequency shift estimation accuracy

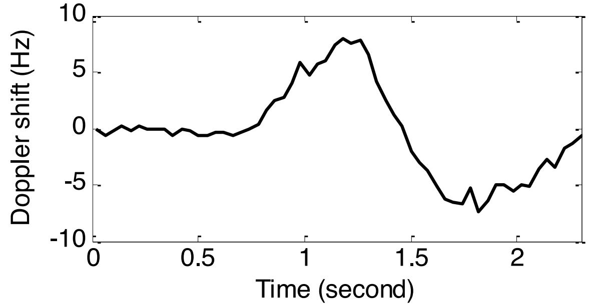
- Utilize larger spectrum
  (Send 5 audio tones in 200 Hz interval)
- Outlier removal
  (Doppler shift change larger than 10 Hz)
- Maximal Ratio Combining



Doppler shift from multiple channels



MRC without outlier removal



MRC with outlier removal

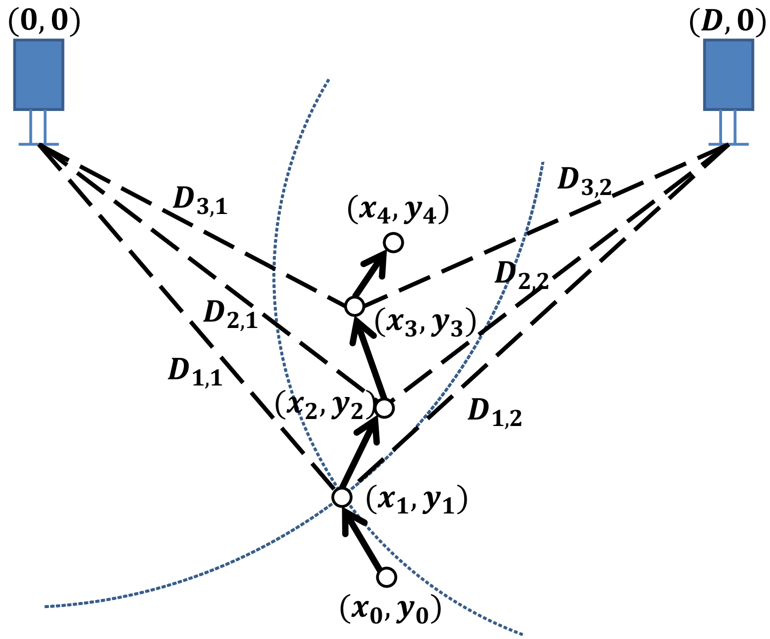
Tracking the device position - 1

- Assumptions
  - The distance between speakers is known
  - The initial device position is known
- Measure Doppler shift from two speakers
- Using the previous position and Doppler, update new distance from speakers

$$D_{i,1} = D_{i-1,1} + \left(\frac{F^{s}_{i,1}}{F_1} c\right) t_s \qquad D_{i,2} = D_{i-1,2} + \left(\frac{F^{s}_{i,2}}{F_2} c\right) t_s$$

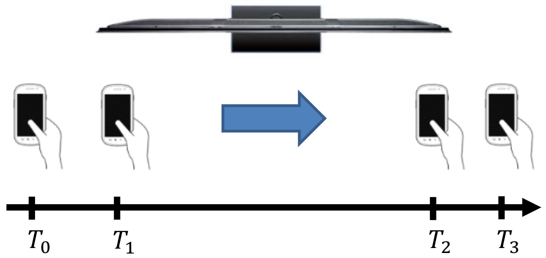
Tracking the device position - 2

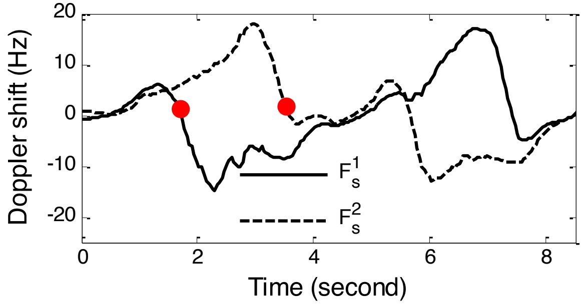
- Having the distance from two speakers, finding the position is finding the intersection of two circles



Finding the distance between speakers

- Calibration to find the exact speaker distance using Doppler
- Moves the speaker from left to right, right to left
  - T_1: F_S^1 changes from positive to negative
  - T_2: F_S^2 changes from positive to negative





Finding the initial position (particle filtering)

- Initially allocate many particles in a given space

$$\boldsymbol{P} = \{(x_o^1, y_o^1), \dots, (x_o^N, y_o^N)\}$$

- In each movement update, remove particles that give infeasible movement (i.e., D < D_1 + D_2)
- Update position by averaging the movement of all remaining particles

$$(x_{i+1}, y_{i+1}) = \left(x_i + \frac{\sum_{k\in\boldsymbol{P}}\left(x_{i+1}^k - x_i^k\right)}{|\boldsymbol{P}|}, y_i + \frac{\sum_{k\in\boldsymbol{P}}\left(y_{i+1}^k - y_i^k\right)}{|\boldsymbol{P}|}\right)$$

Controlling a device with one speaker

- Leverage one speaker and one RF signal (WiFi or Bluetooth)
  - Measure the change of the distance from speaker using Doppler shift
  - Measure the change of the distance from RF source using phase change

$$d_{t2} = \left(\frac{\theta_{t2} - \theta_{t1}}{2\pi}\right)\lambda + d_{t1}$$

d_t: Distance at time t
λ: wavelength of the signal
θ_t: Phase of the received signal at time t

- Use the same mechanism to track the position

Implementation

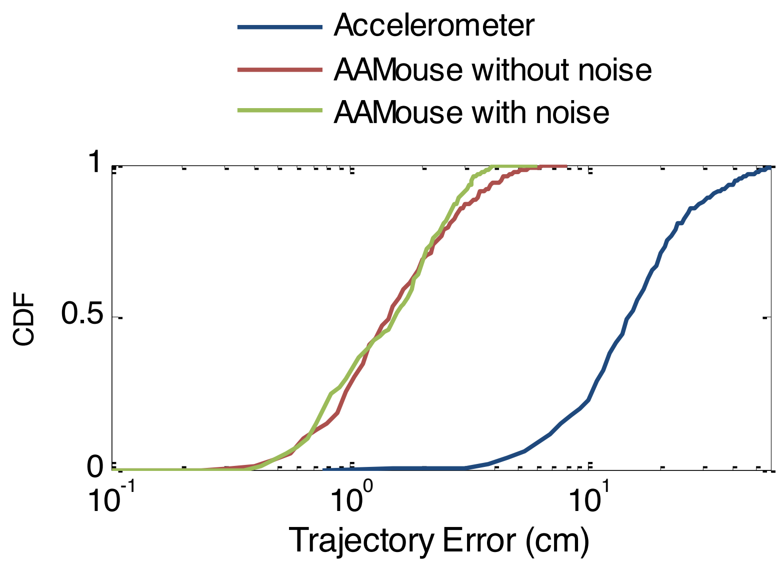
- We implement AAMouse to enable real-time device tracking
  - Mobile application – collect audio signal from the microphone and deliver it to the tracking processor
  - Tracking processor - track the device, visualize the current position
- For comparison, we also implement
  - Camera based tracking (ground-truth)
  - Accelerometer based tracking
  - Gyroscope
    - Moves the pointer by rotational movement of the device
    - Used in commercial air mouse devices

Performance evaluation

- We performed user study with 8 students, and evaluated
  - Tracking accuracy
  - Target pointing
    - Show a point on the screen
    - Ask the user to point it
  - Drawing
    - Show a simple shape such as heart, circle, diamond, triangle
    - Ask the user to draw it

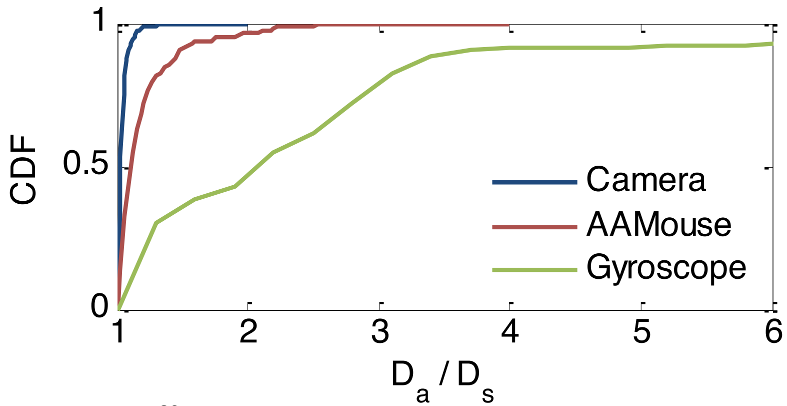
Tracking accuracy

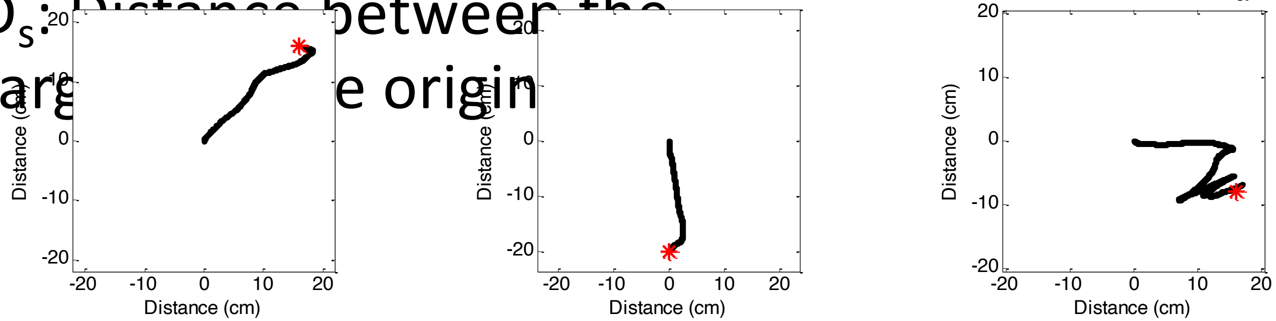
- AAMouse
  - Median error: 1.4 cm
  - 90th percentile error: 3.4 cm
  - Not affected by background noise
- Accelerometer
  - Median error: 17.9 cm
  - 90th percentile error: 37.7 cm



Target Pointing evaluation

- $R = D_a / D_s$
  - D_a: actual distance the pointer travels to reach the target
  - D_s: Distance between the targ … e origin



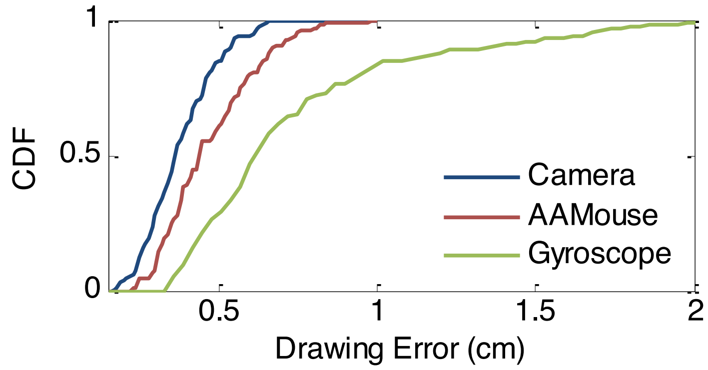


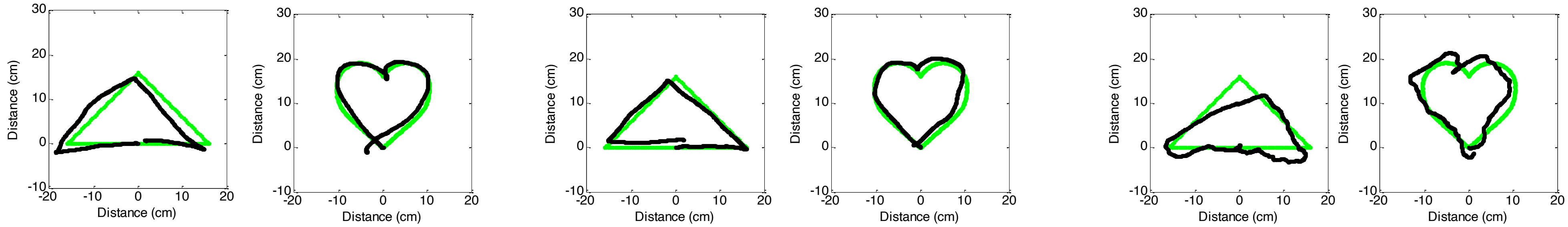
AAMouse Camera Gyroscope

Samples from 80th percentile R

Drawing evaluation

- Drawing error
  - The distance between original shape and the drawn image





AAMouse

Camera

Gyroscope

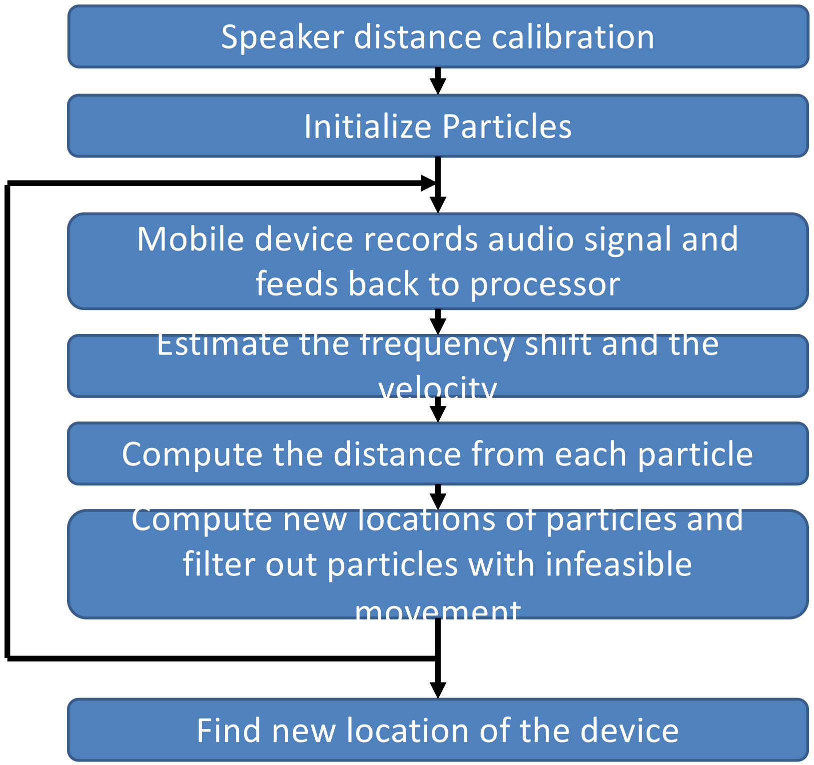
Samples from median drawing error

Conclusion

- We develop an audio based device tracking and apply it to realize mouse
- Future work
  - Improving the accuracy and robustness of the tracking
  - Realizing 3D tracking

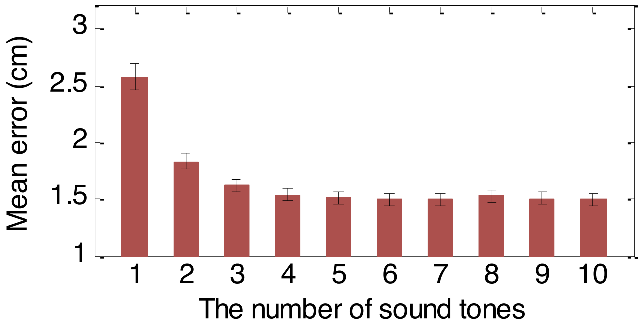
Thank you!

Putting it all together

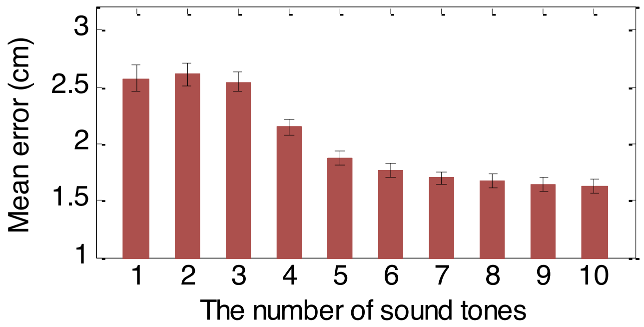


Micro benchmark - 1

- Impact of using multiple sounds and outlier removal



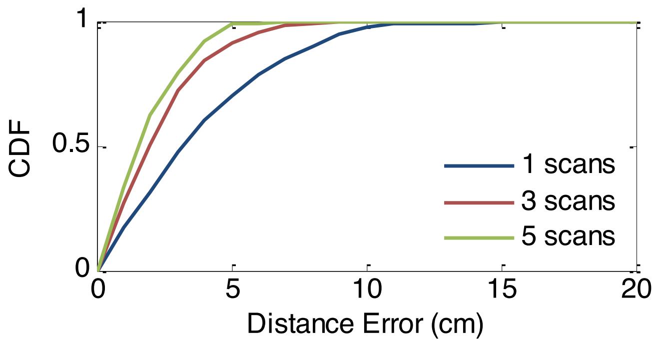
With outlier removal



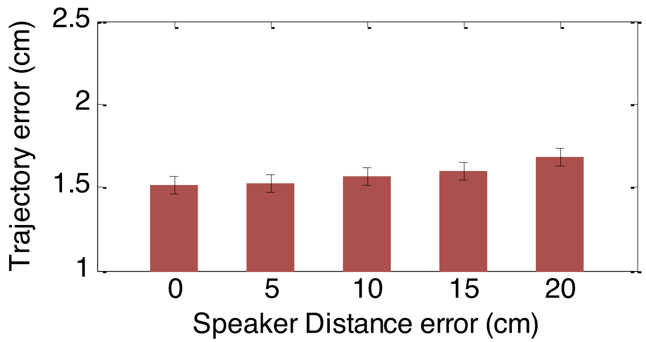
Without outlier removal

Micro benchmark - 2

- Impact of the speaker distance calibration



CDF of the speaker distance error



Trajectory error with various speaker distance errors

Motivation

- Goal: precise device tracking technique to use a mobile device as *Air mouse*
- Applications
  - Remote controller for Smart TV, Google Glass, laptop and PC
  - Motion controller for gaming applications (e.g., X-Box Kinect, Nintendo Wii)
  - Gesture recognition for Smart watch

Accelerometer based device tracking

- Accelerometers are equipped in most mobile devices
- Ideally, we can track the device position by double integration of the acceleration

$$a = \frac{dv}{dt}, \qquad v(t) = \int_0^t a(t)\,dt, \qquad s(t) = \int_0^t v(t)\,dt$$

- In practice, hand vibration significantly affects the acceleration, and small error accumulate quickly due to double integration