Depth Aware Finger Tapping on Virtual Display

Ke Sun[†], Wei Wang[†], Alex X.Liu[†‡], Haipeng Dai[†]
Nanjing University[†], Michigan State University[‡]

Mobisys'18 June 3, 2018

Motivation

Traditional tapping-based interaction:

- Require physical devices
- Limit the freedom of user hands

Motivation

Tapping-in-the-air:

- Hands are free to interact with other objects
- Depth measurements provide different levels of feedbacks

Limitation of Prior Arts

Customized depth-cameras

- Low accuracy:
  Centimeter-level accuracy (without different levels feedback)
- High lantency:
  Low frame rate and high computational requirements

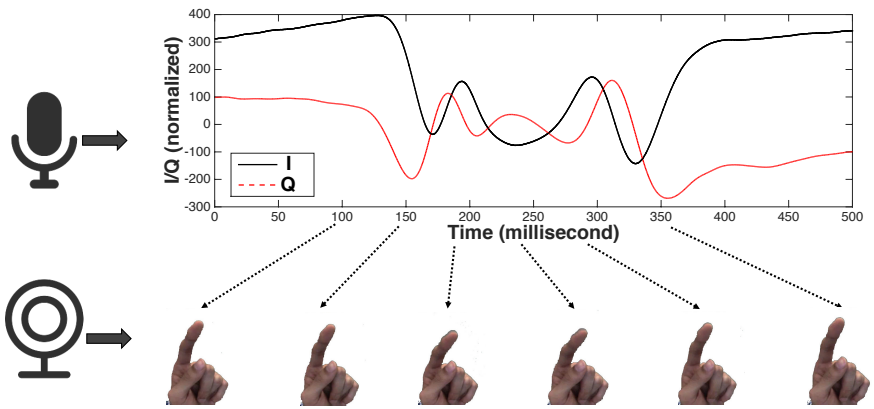
Problem Statement

Can we suppport tapping-in-the-air without depth-cameras?

and meet these design goals

- High accuracy (mm-level)
- Low latency (< 20 ms)
- Different levels feedback (finger bending angle)
- Low computational cost (works on mobile devices)

Basic Idea



Ultrasound

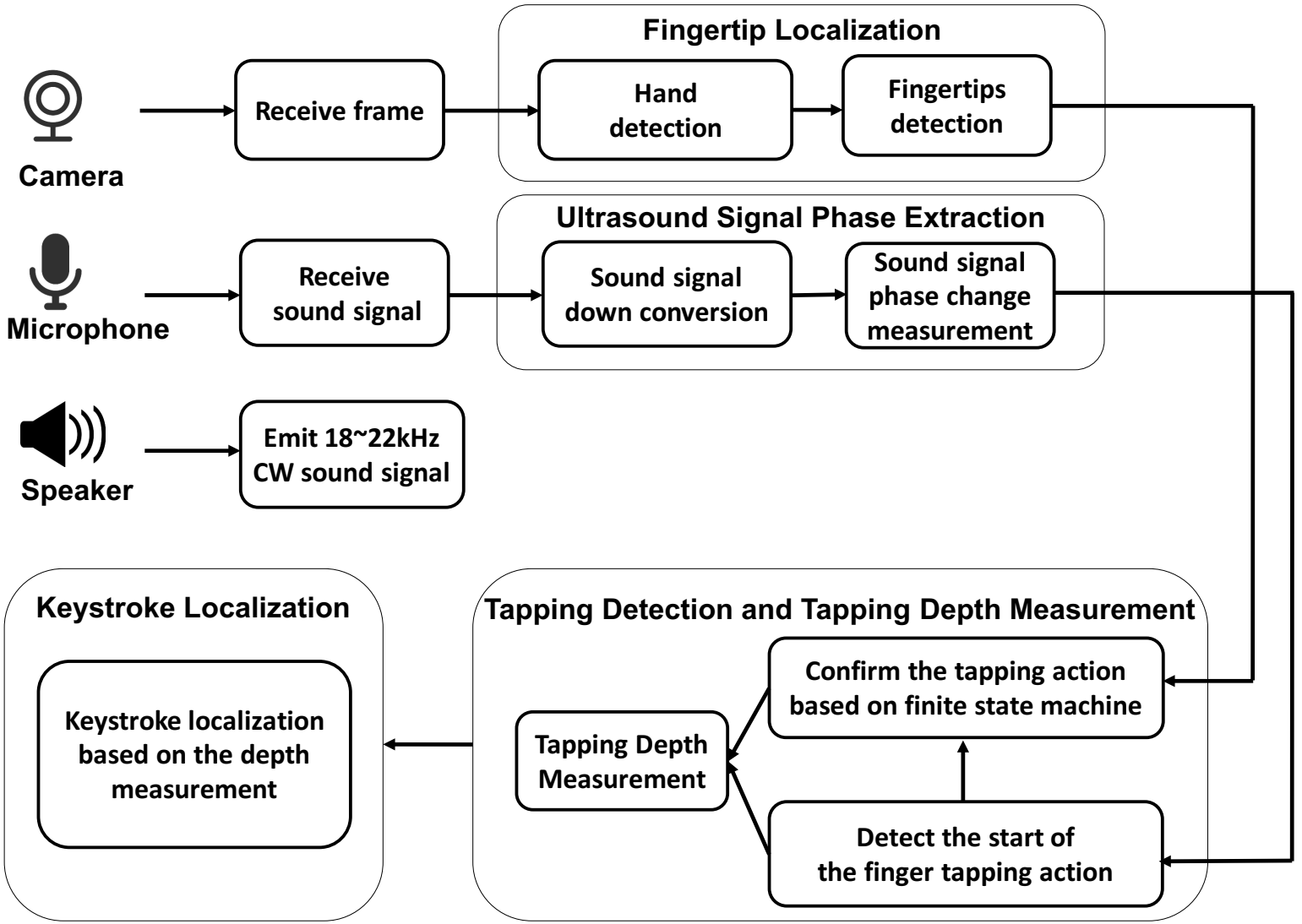
High sampling rate (48 kHz)
Sensitive to the depth direction
Only 1D information

Mono-camera

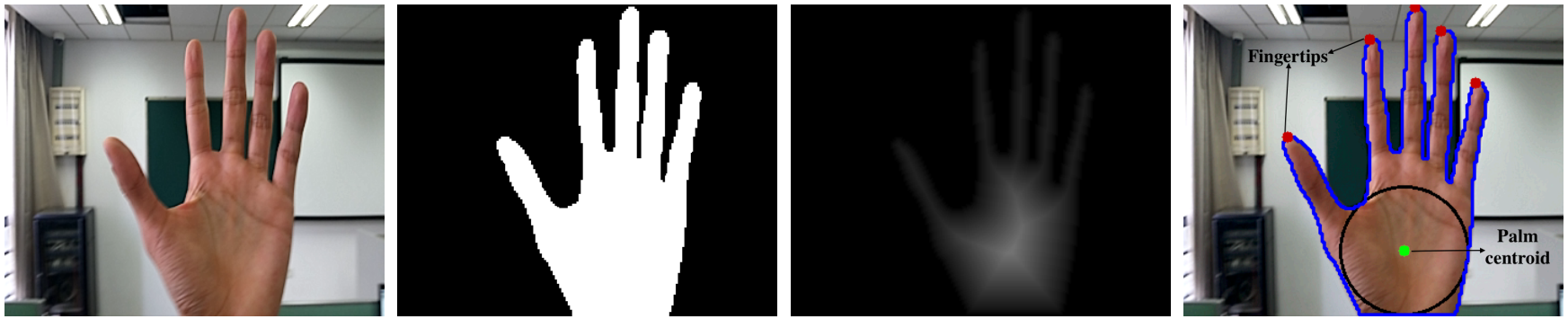
Low frame per second (30 fps)
Accurate 2D information

- Use ultrasound based sensing, along with one COTS mono-camera, to enable 3D tracking of user fingers with high frame rate.

System Architecture



Fingertip Localization

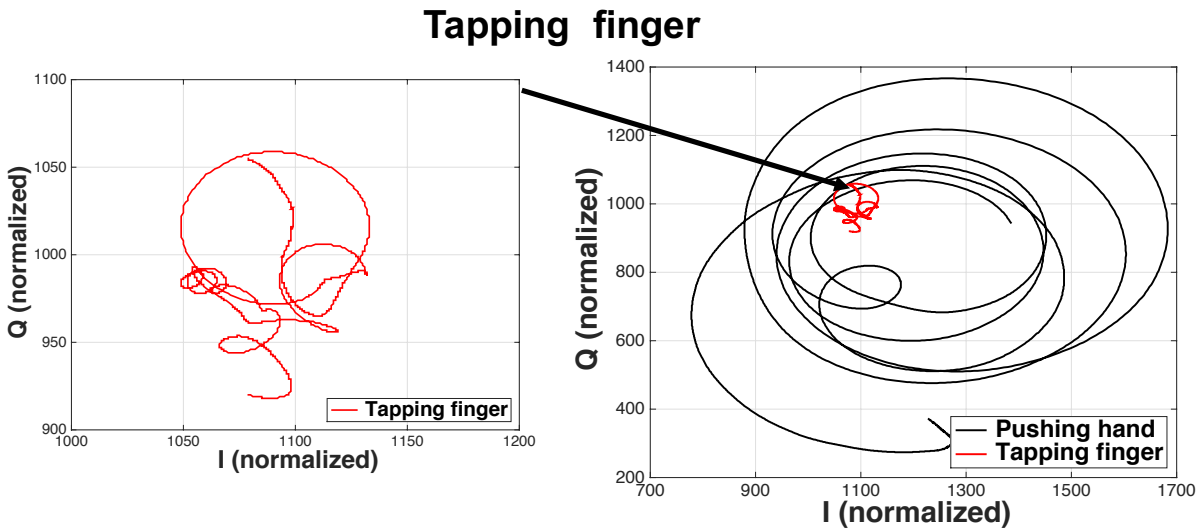


(a) Input frame (b) Binary image (c) Distance transform (d) Fingertips image

Light-weight computer vision algorithm to locate the fingertips in 2D

- Adaptive Skin Segmentation:
  Otsu's method calculates the optimal threshold
- Hand detection
  Find the centroid of the palm (Distance Transform)
- Fingertip Detection for tapping gesture
  Extreme-points-based scheme

Ultrasound Signal Phase Extraction

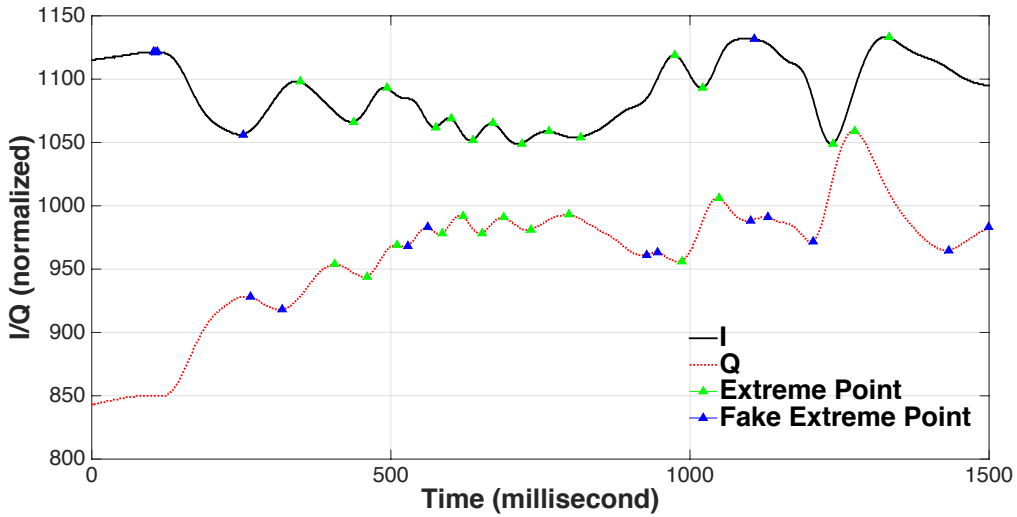


- Phase-based distance measurement
  - Measure phase changes caused by the movement
  - 16 single frequencies ($17 \sim 22$ kHz) linear regression

Challenge:

- Phase changes caused by the finger movements is much smaller.
- Multipath interference in finger movements is much more significant.

Ultrasound Signal Phase Extraction



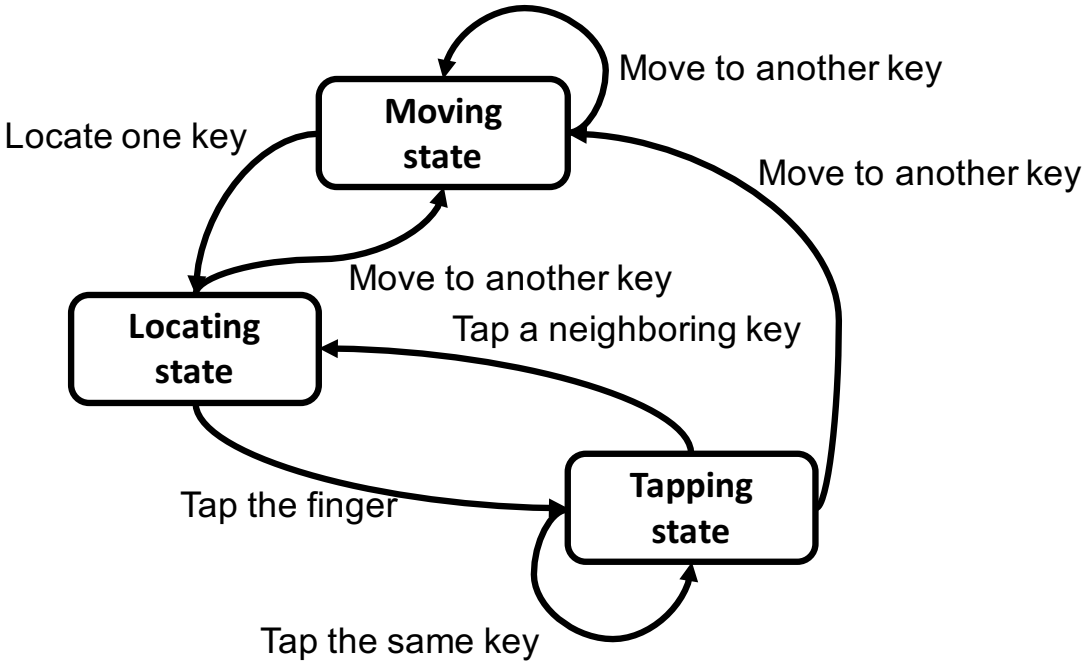
Peak and Valley Estimation

- Find the peak and valley
  - Avoid the error-prone step of static vector estimation
- Exclude the fake extreme points:
  - "FingerInterval": the magnitude gap of the finger
  - "SpeedInterval": the speed of the finger
- Future: use modulated signal to locate the absolute distance and exclude other distance dynamic multipath

Finger Motion State

- "Moving state"–Moves their finger to the key
  - Video: Track the fingers
  - Audio: Difficult to build the model
- "Locating state"–Keeps their finger on the target key position briefly
  - Video: Difficult to perceive
  - Audio: Detect the short pause
- "Tapping state"–"Tapping down state" & "Tapping up state"
  - Video: Difficult to measure
  - Audio: Measure the depth information

Finger Motion Pattern



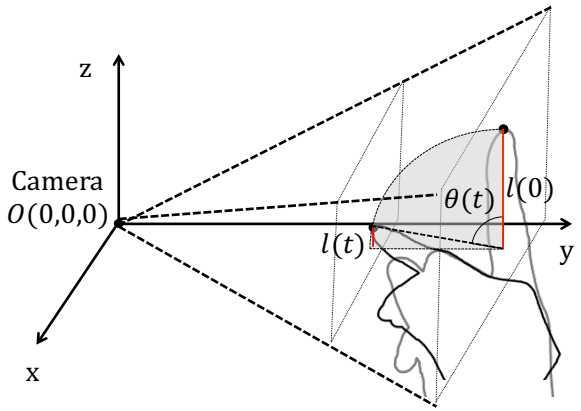
- Tapping a non-neighboring key
  - "Moving state" –> "Locating state" –> "Tapping state"
- Tapping a neighboring key
  - "Locating state" –> "Tapping state"
- Tapping the same key
  - "Tapping state"

Finger Tapping Detection

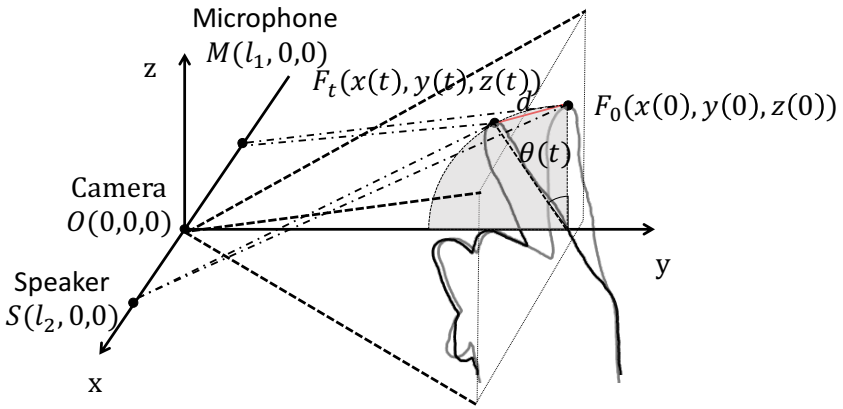
1. Audio to detect that the "tapping state"
   - High sampling rate (48 kHz) –> Low latency
   - Sensitive in the depth direction –> High accuracy
   - High false positive rates
2. Video to look back to the previous frames
   - Measure the duration of "Moving state" and "Locating state"
   - Check the state machine to remove false alarms–> High robustness
   - Measure the depth of finger tapping
3. Keystroke localization
   - Calculate the location of the fingertip during the "Locating state"
   - Determine the fingertip with the largest bending angle
   - 1-NN determine the pressed virtual key

Measure the depth of finger tapping

- Measure the bending angle of the finger
  - Deep finger tapping: camera-based model



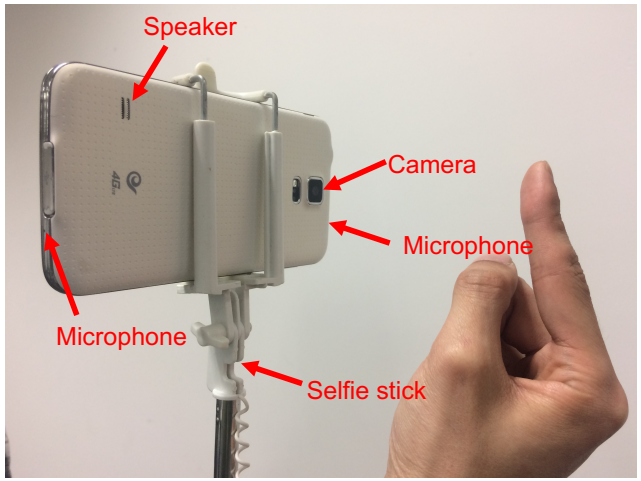
  - Gentle finger tapping: ultrasound-based model



Implementation

Implemented on Android with NDK

- Video: OpenCV C++
- Audio: C++



Video parameters used

24 frame per second
355×288 resolution

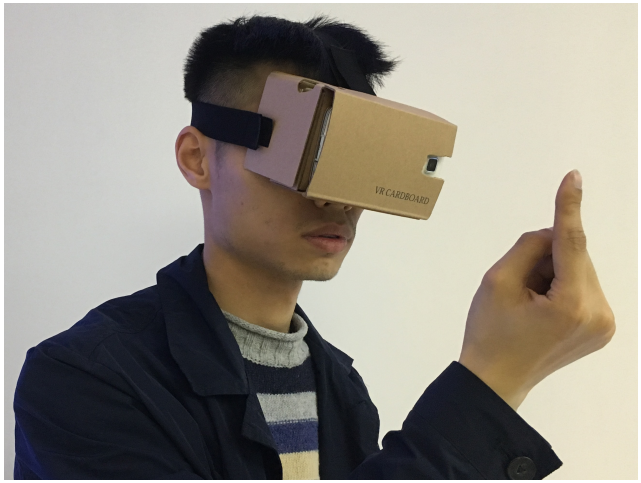
Audio parameters used

48 kHz sampling rate
512 samples per segment (10.7 ms)
16 single frequencies ($17 \sim 22$ kHz)

Evaluation Setup

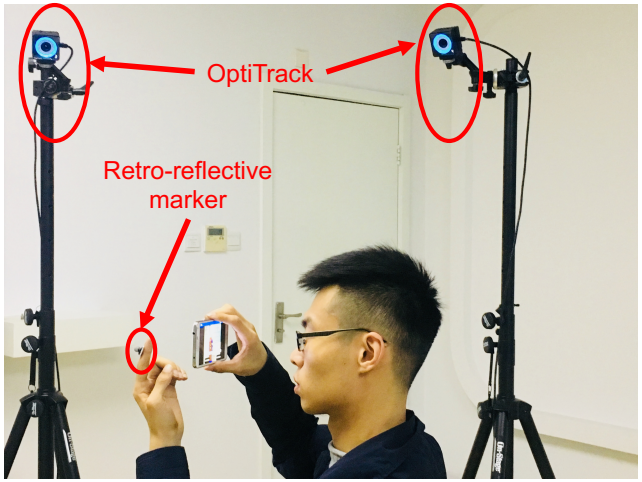
Three different use cases:

- Fix by selfie stick
- Hold in hand
- Set on the head by cardboard VR

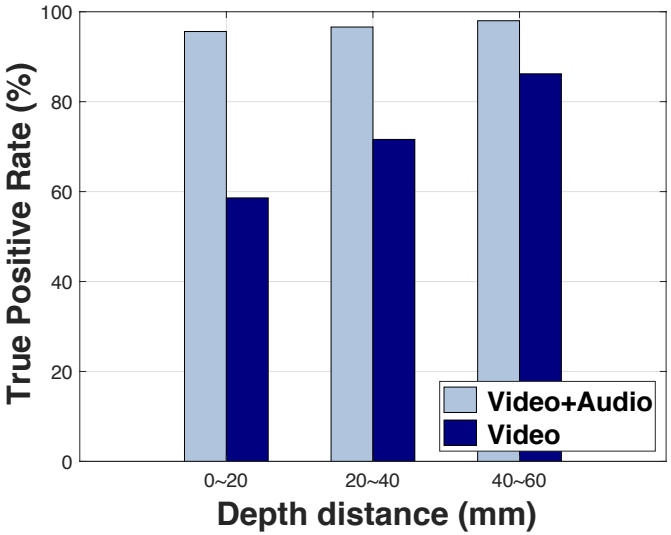


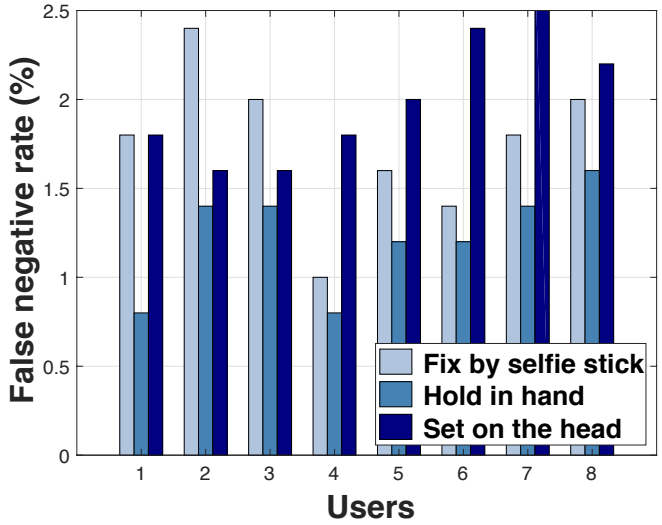
Depth ground truth:

- OptiTrack
  (4 depth cameras + 120 fps)



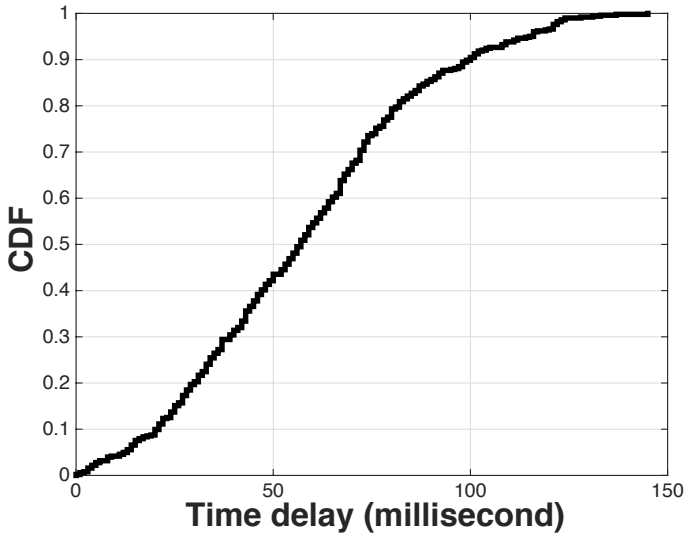
Result – Accuracy





- Average movement distance error of 4.32mm (SD $= 2.21$mm)
- Average 98.4% accuracy with FPR of 1.6% and FNR of 1.4%
- Improve the gentle finger tappings accuracy from 58.2% to 97.6%

Result – Latency



(a) Audio thread

	Down conversion	PVE	Tapping detection	Total
Time	6.455ms	0.315ms	0.036ms	6.806ms

(b) Video thread

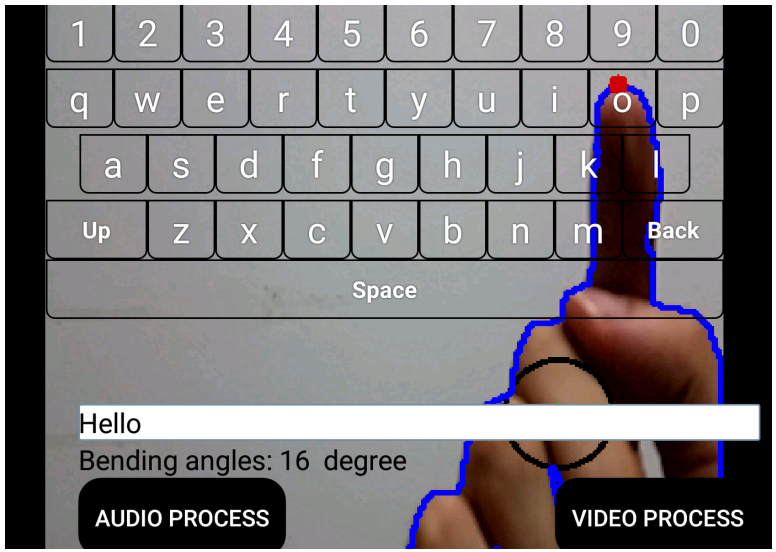
	Hand detection	Fingertip detection	Frame playback	Total
Time	22.931ms	2.540ms	14.593ms	40.064ms

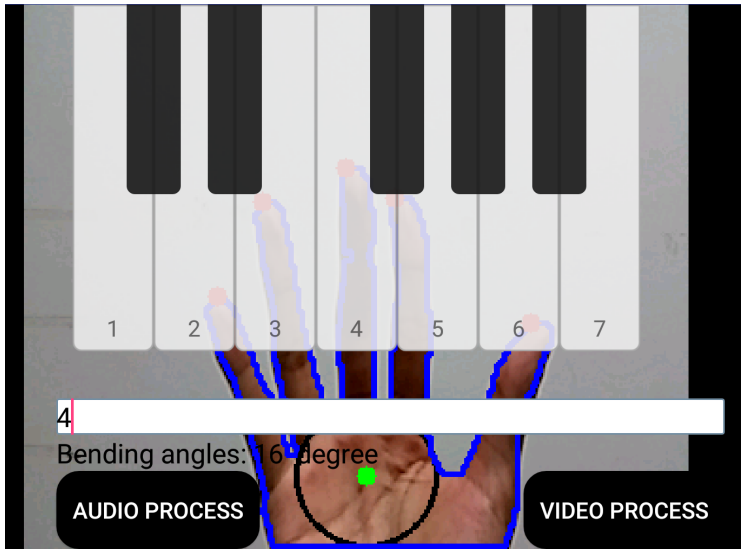
(c) Control thread

	Keystroke localization	Virtual key rendering	Total
Time	0.562ms	10.322ms	10.884ms

- Average response latency of 18.08ms on commercial mobile phones
- Average response latency is 57.7ms smaller than the video-based schemes

Result – Case study





- 12.18 (SD=0.85) WPM for single-finger inputs
- 13.1 (SD=1.2) WPM for multi-finger inputs
- Average 95.0% TPR for 4-level feedbacks

Result – Power consumption

	CPU	LCD	Audio	Total
Idle	$30 \pm 0.2mW$	/	/	$30 \pm 0.2mW$
Backlight	$30 \pm 0.2mW$	$894mW \pm 2.3$	/	$924 \pm 2.0mW$
Video-only	$140 \pm 4.9mW$	$895 \pm 2.2mW$	/	$1035 \pm 4.0mW$
Our scheme	$252 \pm 12.6mW$	$900 \pm 5.7mW$	$384 \pm 2.7mW$	$1536 \pm 11.0mW$

- More than 77% additional power consumption comes from speaker
- Significant power consumption overhead of 48.4%
- Future: reduce the power consumption of the audio system

Conclusion

Combining ultrasound sensing information and vision information to achieve tapping-in-the-air

Our system achieves design goals

- High accuracy
  4.32 mm distance error, 98.4% accuray
- Low latency
  18.08 ms, 4x faster than video-based scheme
- Different levels feedback
  based on different bending angles of finger tappings
- Low computational cost
  works on commercial mobile devices

Q&A

Thank you!

Question?