SwordFight: Enabling a New Class of Phone-to-Phone Action Games on Commodity Phones

Zengbin Zhang¹, David Chu², Xiaomeng Chen³, Thomas Moscibroda⁴

¹Univ. of California Santa Barbara
²Microsoft Research Redmond
³Univ. of Science and Technology of China
⁴Microsoft Research Asia
Mobile Gaming Penetration

US Mobile App Time Consumption

Games 49%
Social Networking 30%
Other 15%
News 6%

Global Mobile Gaming Revenue

Billions

0.00 2.00 4.00 6.00 8.00 10.00 12.00 14.00 16.00 18.00

Typical Mobile Games

Can play it anywhere 😊
Physical actions are VERY limited 😞
Motion Games

Involve more physical actions 😊
Need a well-setup game environment 😞
Mobile + Motion Games (MMG)

A SwordFight with your phones!

Attack  Block
Enabling SwordFight

Game Actions

“How far is my enemy?”

Attack

Block

Communication

Gesture Recognition

REAL-TIME, ACCURATE Phone-to-Phone Distance Measurement

Key Enabler
Distance Measurement in MMG

1. No infrastructure support, commodity phones

2. Real-time measurement
   - Low measurement time: \( \leq 100\text{ms} \)
   - High measurement frequency: > 10 times/second

3. Accurate
   - Centimeter level accuracy

4. Robust
   - Mobility: Up to 4m/s hand-to-hand speed
   - Environment noise: talking, laughing, shouting, walls, ...
Ranging by Sound

• Based on built-in microphone and speaker
  – *e.g.* Beepbeep[\textit{Sensys’07}], Phone-to-Phone 3D Location[\textit{Sensys’11}]
  – Phones send “tones” to each other
  – Compute distance from tones’ arrival time

Accurate (\~2cm error) 😊 Not suitable for MMG 😞
No need for infrastructure 😊 Why?
Not Suitable for MMG

• Assuming phones are static, takes ~1 second to measure distance
  – Hand-to-hand speed up to 4m/s in motion games
  ➔ Measurement error is in meters!
  ➔ Low frequency!

• Doppler Effect happens at mobile scenarios
  – Received tones will be perturbed
  – Tone detection algorithm will not work
Our Proposed Design

Real-time measurement

- Real-time Tone Detection Algorithm

Pipelined Streaming Measurement Structure

Adaptive Parallel Tone Detector

Robust to Mobility

- Robust to Noise, multipath

Binary Coding, Filtering, etc.

Practical Phone-to-Phone Ranging System for MMG
Outline

• Motivation

• Fast, Accurate, Robust Ranging for MMG
  – Real-time distance measurement
  – Robust to mobility

• Evaluation

• Conclusion
Typical Acoustic Ranging Process

Phone A
Phone A's recording stream

Phone B
Phone B's recording stream

Locate by tone detection algorithm

\[
\text{Distance} \approx \frac{(t_{A2} - t_{A1}) - (t_{B2} - t_{B1})}{2} \times v_{\text{sound}}
\]
Why Real-time is Hard

MIC initialization lag
200~600ms

Recording
~100ms

Buffering
0~50ms

Distance Calculation
1~6ms

Audio playing lag
~100ms

Phone A’s tone
~50ms

Phone B’s tone
~50ms

Total measurement time ~1s
Our Approaches

• Streamlining tone sending & recording
  – To eliminate hardware lags

• Pipelining recording, tone detection & distance calculation
  – To increase measurement frequency
  – Tone exchange and detection are in parallel

• Design a real time tone detection algorithm
  – To fundamentally reduce measurement delay
Measurement Structure

Threads:

Distance Calculation

Tone Detection

Recording

Tone Sending

Phone A  Phone B  Phone A  Phone B  Phone A
Measurement Structure

Threads:
- Distance Calculation
- Tone Detection
- Recording
- Tone Sending

Sound

Ambient Noise Cancellation

Autocorrelation(s)

Smoothing

Multipath Filter

Tone detected?

yes

Cross-correlation

Measurement Exchange

Distance Calculation

Low Pass Filter

<distance values>

Doppler Predictor

WiFi
**Existing Tone Detection Algorithm**

Cross-correlation

Cross-correlation Peak → Arrival Time

Recorded Sound

Template

$X_i$  

$T_i$

Correlation Window

Each window takes $O(W)$ time  
*e.g.* $W=512$ sound samples in our system

100ms sound stream = 200ms computation!  
Not possible to have real time detection!
Autocorrelation-based Detection

Autocorrelation

Autocorrelation Peak

Actual Location

Autocorrelation peak is flat. Error can be 20cm+. 

Recorded Sound

Delayed Sound

$X_i$  $Y_i$

Computed in $O(1)$ time from previous window, much smaller than $O(W)$

100ms sound stream = 20ms computation!
How to Maintain Accuracy

- Cross-correlation peak is sharp, accurate, close to autocorrelation peak!
- Solution:
  - Set a “search window” around autocorrelation peak
  - Apply cross-correlation in search window

→ 100ms sound stream = 60ms computation!
Example

Tones

Autocorrelation results
Not very smooth

Auto Correlation

After smoothing autocorrelation peaks

Detected cross correlation peaks

Ground truth of cross correlation peaks

Cross correlation in a small window

Smoothing
Outline

• Motivation

• Fast, Accurate, Robust Ranging for MMG
  – Real-time distance measurement
  – Robust to mobility

• Evaluation

• Conclusion
Impact of Doppler Effect

• Example: approaching ambulance has higher pitch
  – Sound wave is “squeezed”

Length of recorded tones are changed

Tone length $W \rightarrow W'$, e.g. $W' = W + 1$
Impact of Doppler Effect

- **Impact**
  - Should adapt to *new tone length* $W'$ in Autocorrelation

- **Q: How to compute $W'$?**
  - Theoretically, by two phones’ speed: Unknown 😞
  - Solution: max hand speed is known $\rightarrow W'$ is in $[W-2, W+2]$
Addressing Doppler Effect

• Strawman solution
  • 5 parallel detectors to search in $[W-2, W+2]$

• Adaptive solution
  – Reduce to 3 parallel detectors, ~90ms to compute 100ms sound
Additional System Components

• Use high pass filter to handle ambient noise

• **Binary coding**: two phones use different beep sounds
  – A missing beep can cause successive errors

• Use **Kalman filter** to correct errors & smooth results

• Use techniques to address multipath reflection

• **Recovery** mechanisms if phones get “out-of-sync”
Implementation & Evaluation

• Two games
  – Sword Fight, Chase Cat

• Two types of platforms
  – Nexus One (Android)
  – Samsung Focus (Windows Phone 7)

• Evaluation
  – Controlled experiments
  – Gaming in the wild: 400+ players, demonstrated at MS Techfest
Performance Summary

• Measurement frequency: 12Hz

• Measurement lag: around 100ms

• Accuracy: 2cm median error

• Tone detection under Doppler effect: 95% (2m/s), 86% (4m/s).

• Works well under noise of the same power level
Comparing with Kinect

- 2 players play SwordFight in front of a Kinect sensor
Conclusion

• We design and implement Mobile Motion Games
  – Incorporate physical actions, play wherever you like

• Enabled by real-time, accurate distance measurement
  – Demonstrated by two games on two platforms

• Comparison to *kinect* shows similar performance