KOREA UNIVERSITY

BatTracker: High Precision Infrastructure-free Mobile Device Tracking in Indoor Environments

Sensys 2017

Bing Zhou, Mohammed Elbadry, Ruipeng Gao, Fan Ye

2018.11.06

박성수

1. INTRODUCTION

✓ Motivation

• Tracking the continuous movements thus locations of a device in an indoor space is a form of human-computer interaction, popular in many applications including VR/AR, video games and health rehabilitation.



• Using inertial(관성) sensors is the most straightforward approach.
=>However, they suffer from large drift errors over time.

• The state of the art VR systems (e.g., HTC vive [1], Oculus [5]) rely on multiple base stations or cameras to track users in a small area.
=>They increase the cost by several hundred dollars and require installation efforts.



• The Microsoft Kinect [3], leverages a depth sensor with a range of 0.8 − 4m.
=>However, they require a visually distinct target, and their performance is subject to lighting conditions. Besides, cameras may raise privacy concerns.



KOREA UNIVERSITY

✓ Motivation

• RF signals such as Wi-Fi has been widely used for tracking,
=>however they usually require customized hardware and have a limited accuracy due to the high RF signal propagation speed.

• Some recent work has leveraged acoustics for high-precision device tracking by estimating the distances to multiple anchor points.
=>They still require additional infrastructure (e.g., multiple external speakers), which increases the cost and adds configuration efforts.
=>Even though these speakers may be available in users' environment (e.g., speakers on a laptop, TV), they are usually not separated in ideal locations to serve as anchor points.

• In this paper, we propose BatTracker, the first high precision, infrastructure-free mobile device tracking system in 3D space with a range comparable to existing commercial solutions.

KOREA UNIVERSITY

✓ Challenges

• Despite such benefits, accurate, robust and infrastructure-free device tracking based on echoes is far from straightforward.

• Due to the existence of multiple surrounding objects, many echoes, not just those bouncing off objects of large surfaces, but also those from smaller objects or over multiple surfaces (i.e, multi-path), will be received.

• The device movement may create occlusions (by the device, other objects or human body) of the path, thus frequent missing of echoes from reference objects; it can also produce significant noises thus inaccurate acoustic measurements.

• We must design robust algorithms to reliably associate distance measurements to reference objects despite noisy data and frequent missing of desired echoes.

KOREA UNIVERSITY

✓ Contributions

• We achieve high-precision device tracking in 3D indoor environments by incorporating complementary inertial and acoustics measurements, while eliminating the requirement of any additional hardware.

• We design an acoustic sensing technique that can produce accurate distance measurements from the device to nearby objects using echoes, and infer the device's movement velocity from Doppler shifts.

• We follow a multi-hypothesis tracking framework, and design a probabilistic tracking algorithm that fuses distance measurements, Doppler shifts and echo, thus achieving robust and accurate tracking.

• demonstrating that BatTracker can track mobile devices in a typical room with a maximum error of < 1cm for 2D tracking, and a 90-percentile error of 1cm for 3D.

• To the best of our knowledge, BatTracker is the first work to track mobile devices in indoor environments at sub-cm accuracy without the requirement of any additional infrastructure.

2. BACKGROUND

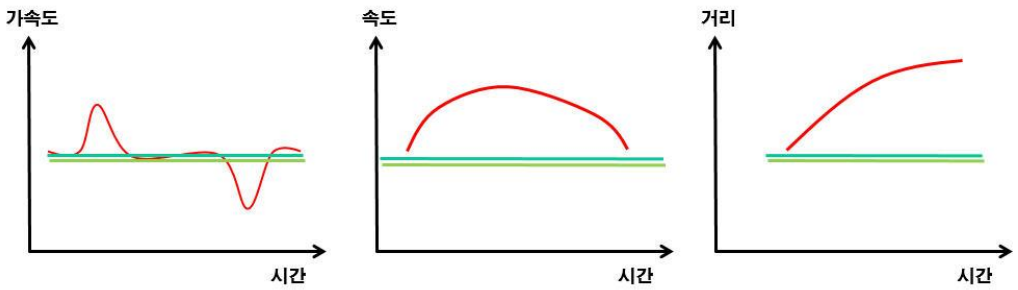
✓ Inertial sensors

• The most straightforward way for tracking is based on inertial sensors.

• Accelerometer and gyroscope embedded in mobile devices allow us to calculate both the direction and speed of the movement.

• Theoretically, we can integrate acceleration to get velocity, and integrate velocity to get the moving distance.

• However, this approach suffers from large drifts over time: the errors can accumulate to meter level in a few seconds.



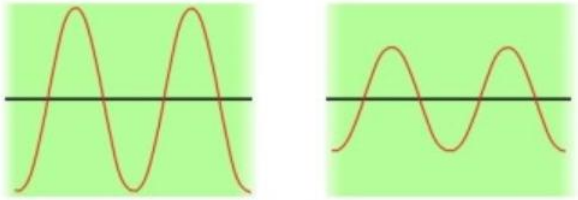
KOREA UNIVERSITY

✓ Three factors of sound

소리의 세기(강, 약) (Amplitude)

진폭에 따라 결정

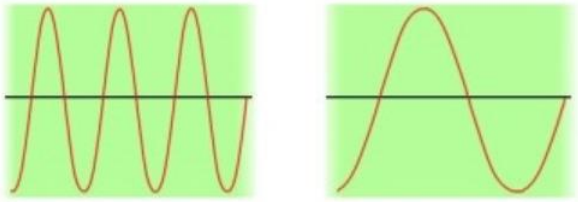
소리의 세기는 진폭의 제곱에 비례



소리의 높이(고, 저) (Frequency)

진동수에 따라 결정

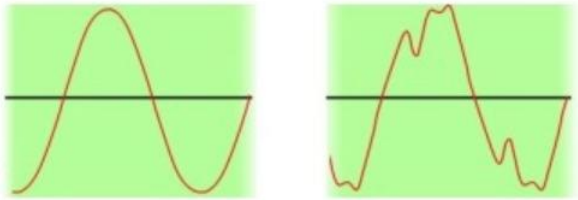
진동수가 많을수록 고음



소리의 맵시(음색) (Waveform)

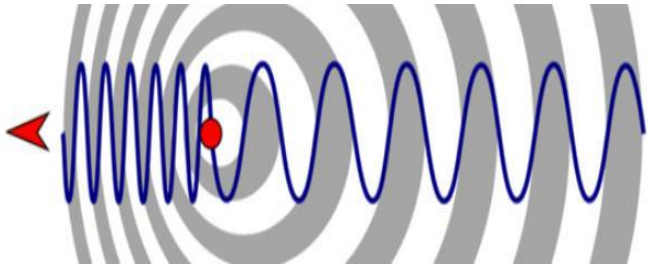
파형에 따라 결정

같은 세기와 높이의 소리라도 구분 가능



✓ Doppler shift

• 파(음파, 전자파)의 발생원과 관측자 간의 상대적인 속도에 따라 파의 주파수가 다르게 측정되는 현상



•구급차가 다가올 때와 떠나갈 때의 알림음이 다름

• 스피드 측정기의 원리

• 전투기 음속(347m/s) 비행시(Sonic Boom)

$$f' = f \times \frac{V - v_o}{V - v_s}$$

위 공식에서

f : 발생원에서 발생시킨 파의 주파수
V : 파장 속도
Vo : 관측자 움직이는 속도
Vs : 발생원의 움직이는 속도





✓ Acoustics

• Acoustics is a favorable sensing modality for ranging and tracking due to the slow sound propagation speed, hence higher accuracy.

• Recent acoustic based device tracking work estimates distances from the device to multiple anchor points, either by integrating the device moving velocity, or directly from Frequency Modulated Continuous Waveform (FMCW) [31] or propagation delay, then triangulate the device location.

• Existing work [40] leverages velocity estimated from Doppler shift for tracking. However, the error accumulation is still non-negligible(무시할 수 없는) over slightly longer time period.

✓ Direct distance measurement

• Direct distance measurement does not require integration hence eliminating error accumulation.

• However, existing work (e.g., CAT [19]) requires infrastructure (e.g., multiple external speakers) as anchor points, which adds to the costs and installation/configuration efforts.

• We decide to leverage echoes from nearby objects with large surfaces (e.g., walls, ceiling/floor, large furniture) commonly existing indoors.

• BatTracker predicts the device position using inertial data in a short recent time window, and immediately correct accumulated error using acoustic measurements.

3. CHALLENGES

KOREA UNIVERSITY

✓ multiple challenges

(1) multi-path effects and clutters in a room make acoustic echoes thus measurements inevitably noisy.

(2) the echo-object association, i.e., which relative distance/velocity measurements correspond to which objects, must be reliably identified.

(3) multiple noisy and error-prone inputs (including inertial, distance and velocity measurements) must be fused effectively and efficiently for robust, accurate tracking.

KOREA UNIVERSITY

- ✓ **four problems**

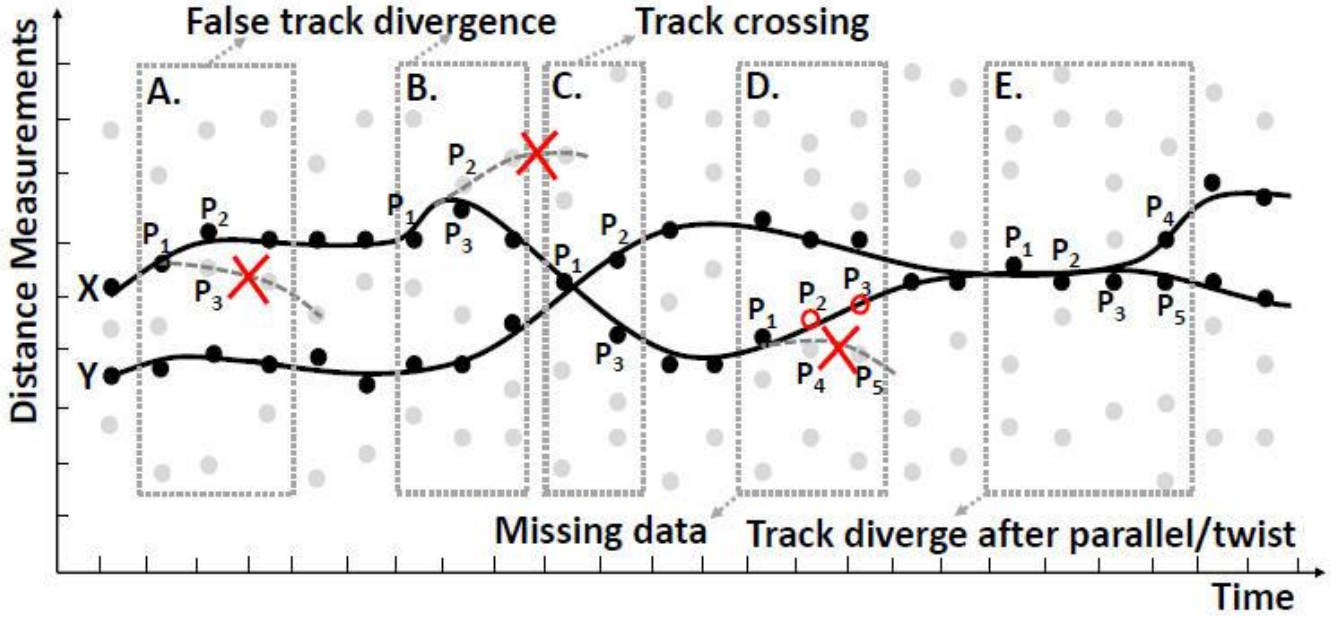


Figure 1: Solid lines represent tracks to two reference objects. Dots at each time stamp are distance measurements from acoustics, where grey ones are from clutters or noise and black ones are from reference objects. Red circles represent missing measurements.

4. BATTRACKER DESIGN

• BatTracker incorporates two sensing modalities for accurate, robust, and infrastructure-free 3D tracking: inertial sensors in a Motion Model for track prediction and acoustics in an Observation Model for track correction (Figure 2).

• Before tracking starts, a trace is initialized by locating the device in the room coordinate system using distances to reference objects.

• Based on the motion model, we predict the position of the device in the next time slot by double integration of acceleration.

• Multiple hypotheses on the association between distance measurements and reference objects, thus the device location, are derived by incorporating Doppler shifts and echo amplitudes.

• The observation model evaluates the strengths of evidences for each of these device location hypotheses, and makes track splitting/pruning decisions to correct errors.

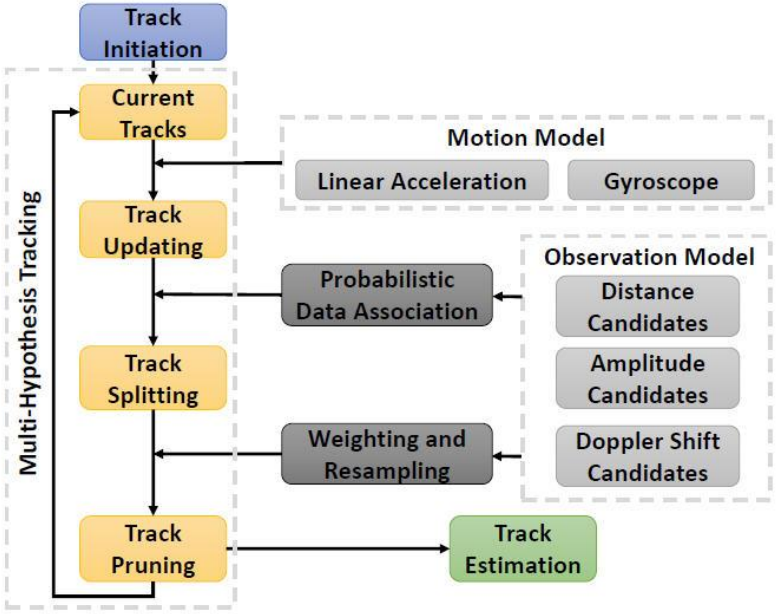
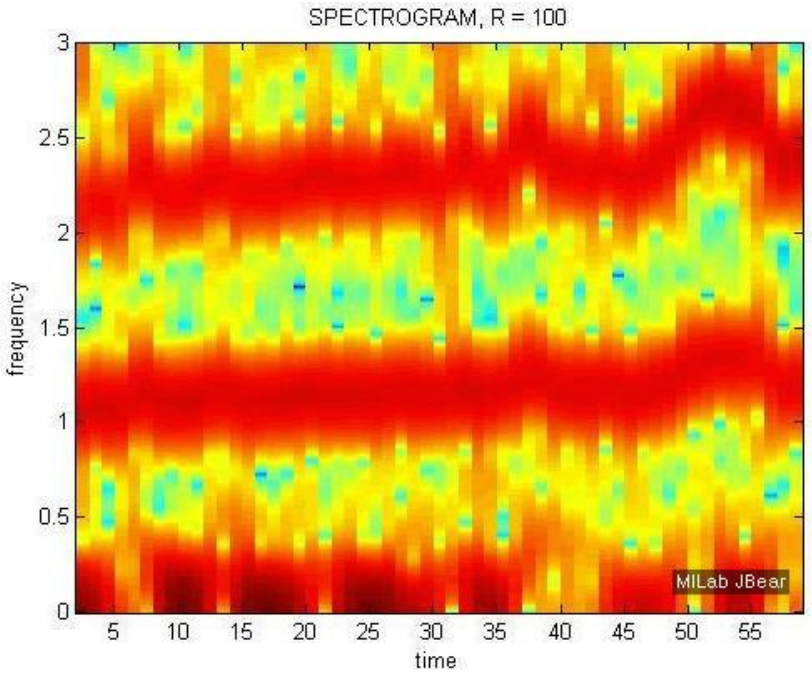


Figure 2: BatTracker uses inertial data in a motion model for track prediction, acoustics in an observation model for track correction, and fuse them into a multi-hypothesis tracking framework for robust tracking.

✓ 4.1 Acoustic Sensing

• The acoustic sensing module of BatTracker consists of signal emitting, recording, and a series of signal processing steps to produce distances, amplitudes and Doppler shifts, hence velocity measurements for echo candidates from nearby objects in indoor environment.

• Unlike some existing work [10, 17] that only produces ranging measurements, we develop signal processing techniques for accurate and reliable range measurement, and additionally Doppler frequency shift, thus both relative distance and velocity to each object.



STFT(Short Time Fourier Transform)
: 시간대별 window에 따른 DFT 추정

✓ 4.1.1 Emitting Signal Design

• we choose signals with a constant frequency, chosen at 17KHz, which is slightly audible to human.

• By setting a moderate volume, our designed sound signal is nearly inaudible to most users, especially under background noise (e.g., music, video games).

• We choose a pulse length of 1ms. Such a short pulse length reduces potential overlapping between echoes traveling similar distances, thus improving the distance measurement resolution.

• Thus the minimum delay between two pulses is $5m \times 2/343m/s \approx 29.15ms$, where 343m/s is the sound propagation speed under a typical room temperature of 25◦C.

• We give a bit buffer space and set it at 30ms.

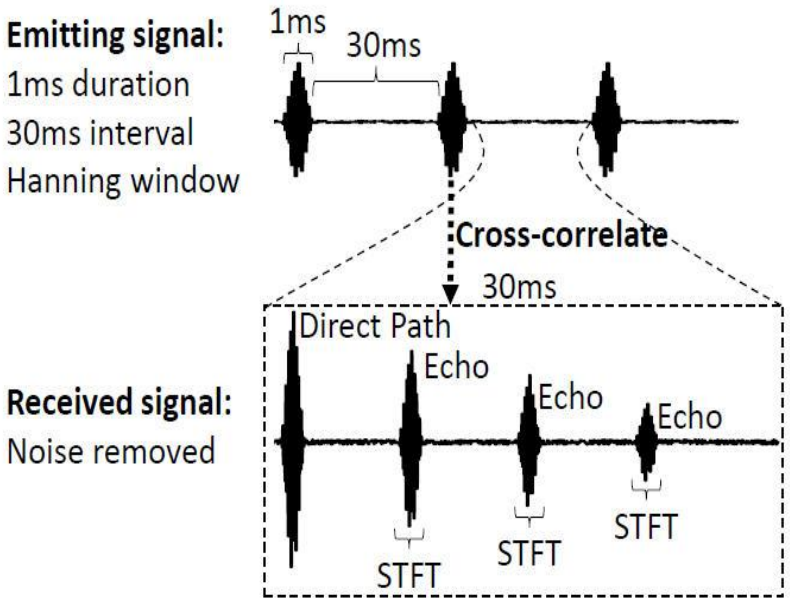


Figure 3: A particular sound signal and multiple signal processing steps produce echo distance, amplitude and velocity measurements.

✓ 4.1.2 Acoustic Measurements Generation

• We develop several steps to generate distance and velocity candidates from received signals.

• Noise Removal : The received signal will go through a Butterworth bandpass filter with pass band of 17K ± 200Hz to remove background noise, while preserving the frequency shift caused by the Doppler effect.

• Locate Each Pulse : We chop the envelop into segments of small time windows of 31ms, each containing echoes from one pulse only. Since the first peak will always be the direct sound from the speaker to the microphone and has the highest amplitude, they are used as the starting points.

• Distance Estimation : Using a threshold, we can select only the top-K strongest peaks, which are hopefully from larger, closer objects. By calculating the time delay between each echo and the starting point, we can estimate the distances between the device and surrounding objects.

• Doppler Shift Estimation : We use a similar approach of artificial padding of zero-valued points described in [40], and apply Short Term Fourier Transform (STFT) to get 1Hz resolution. fd = fe − f is the Doppler frequency shift, f = 17KHz is the frequency of the designed signal, c is the sound propagation speed. we do not use it directly for device moving velocity estimation, but incorporate it as a complementary input for data association improvement.

$$v = \frac{f_d}{2 \cdot f} \cdot c \qquad (1)$$

5. TRACKING ALGORITHM

KOREA UNIVERSITY

✓ Tracking Algorithm

• We design algorithm for track initiation through simple movement gestures, and a multi-hypothesis particle filtering framework for track updating.

• Inertial data serves as motion model for state prediction, acoustic distance measurements are observations for state correction.

• Echo amplitudes and velocities from Doppler shifts are leveraged for data association problem and importance weight estimation.

✓ 5.1 Track Initiation

• Track initiation is the process of finding major reference objects with stable echo reflections, and determining the distances to each of them, such that we can track such distances continuously.

• In practise, two adjacent side walls and the ceiling can be a very good reference object combination as they are relatively large and clean.

• Figure 4 shows the reference coordinate and the device local coordinate, the x-y plane is the ceiling, y-z and x-z planes are two perpendicular side walls.

• There are two major challenges for track initiation:

i) how do we find out objects that create strong and stable echoes, hence with less chance of data missing.

ii) how to select three stable objects, and associate each of them to the x, y, z directions in reference coordinate.

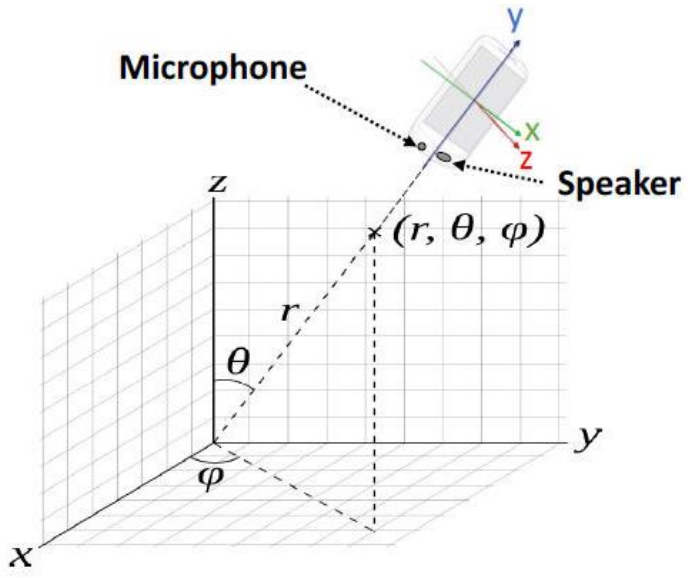


Figure 4: Microphone and speaker are at the bottom of the mobile device, which is pointed to a corner in a room for reliable echo detection.

✓ 5.1 Track Initiation

• We solve the above problem by simple movements using a statistic approach.

• Without loss of generality, we illustrate the process of finding reference object (e.g., x-z plane) in Figure 5.

• We perform the following two steps, track candidate generation and track association for initiation.

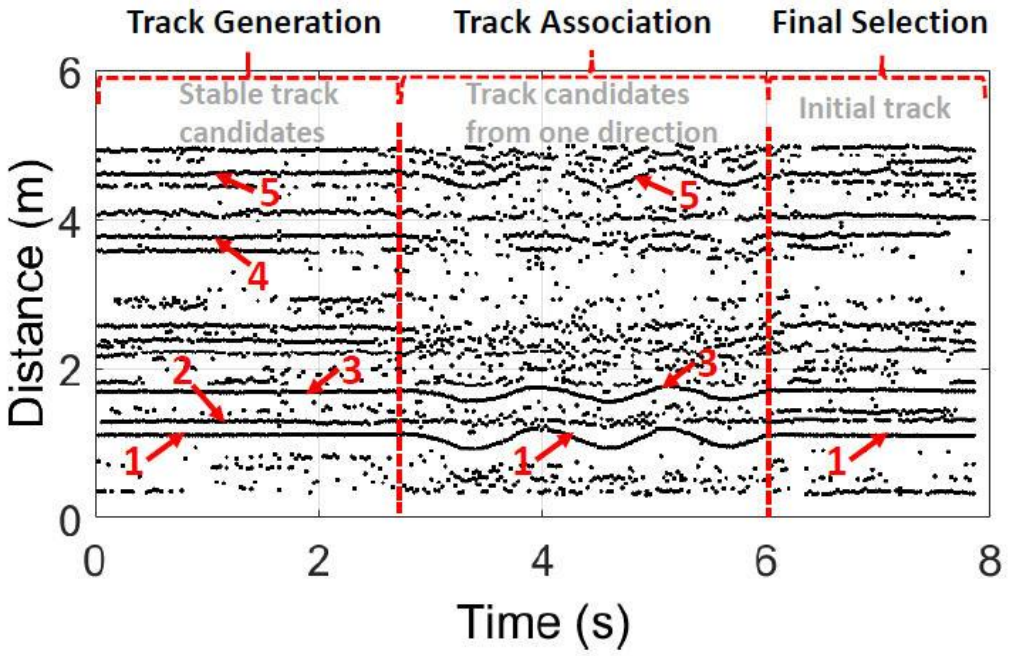


Figure 5: Track initiation process in one direction, which consists of track candidate generation, track association and final track selection.

✓ 5.1.1 Track candidate generation

• We point the phone's bottom (where speaker and microphone locate) to one direction (e.g., perpendicular to x-z plane), hold the phone still for a few seconds (0 − 2.7s in Figure 5).

• The horizontal line between 0 − 2.7s corresponds to nearby objects from multiple directions.

• We use a simple density based clustering approach to locate these candidates, and keep the top-K candidates with least measurement variance as stable potential track candidates.

• In this example, we can easily notice that there are 5 stable candidates as marked in Figure 5.

• we still need to figure out which one is from x-z plane along y axis.

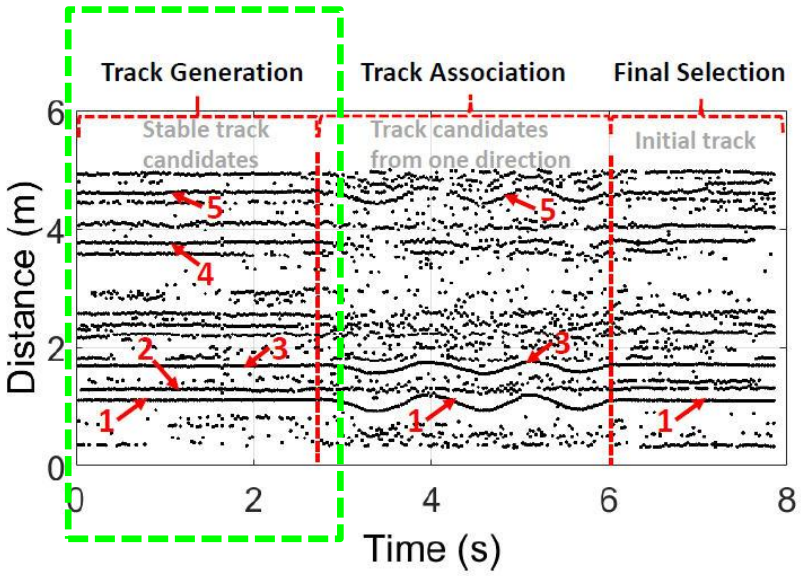


Figure 5: Track initiation process in one direction, which consists of track candidate generation, track association and final track selection.

✓ 5.1.1 Track association

• We make minute movement back and forward along y axis several times (2.7 − 6s in Figure 5).

• During this moving period, echoes coming along y axis shows a strong robustness to noise (e.g., track candidate 1, 3, 5).

• We derive the moving acceleration by double deviation of the distance measurements, and crosscorrelate this derived acceleration with the one measured from inertial sensor to get the similarity.

• Now we know track 1, 3, 5 are coming along y axis, and we select the one with highest cross-correlation similarity (e.g, track 1) as our final initial track.

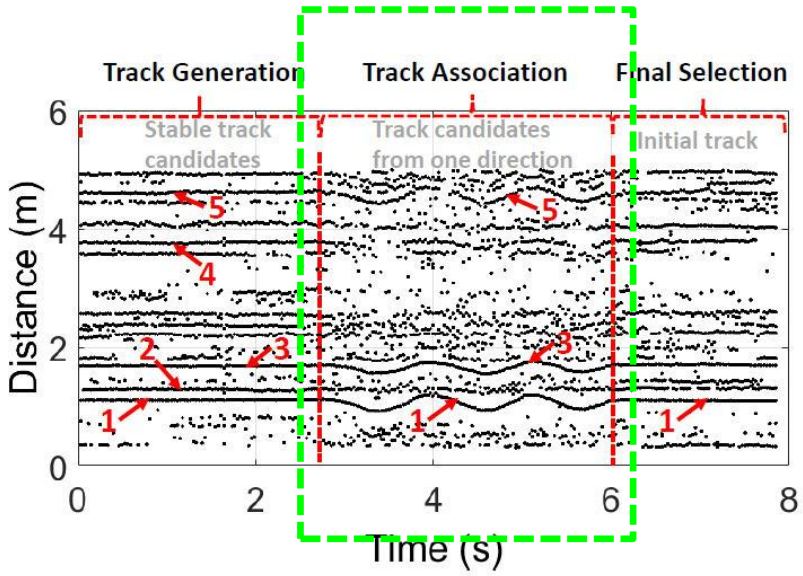


Figure 5: Track initiation process in one direction, which consists of track candidate generation, track association and final track selection.

✓ 5.2 Track updating

• We update the track over time based on the sensing data from inertial sensors and acoustics, where inertial data serves as motion data for device position prediction, and acoustic measurements serve as observations for position correction.

• We start from introducing the motion model and observation model in our design, then present how the challenges are solved using a probabilistic Multi Hypothesis Tracking (MHT) framework.

• MHT has been used in computer vision and radar applications for human or aircrafts tracking [24, 29].

•MHT allows a track to be updated by more than one measurement at each update, spawning multiple possible tracks, and it calculates the probability of each track and typically only reports the most probable of all the tracks.

• Extended Kalman Filter (EKF) and particle filters are commonly used in MHT problems [12, 14].

KOREA UNIVERSITY

✓ 5.2.1. Models

• Following the MHT framework, we design the motion model and observation model for our tracking system.

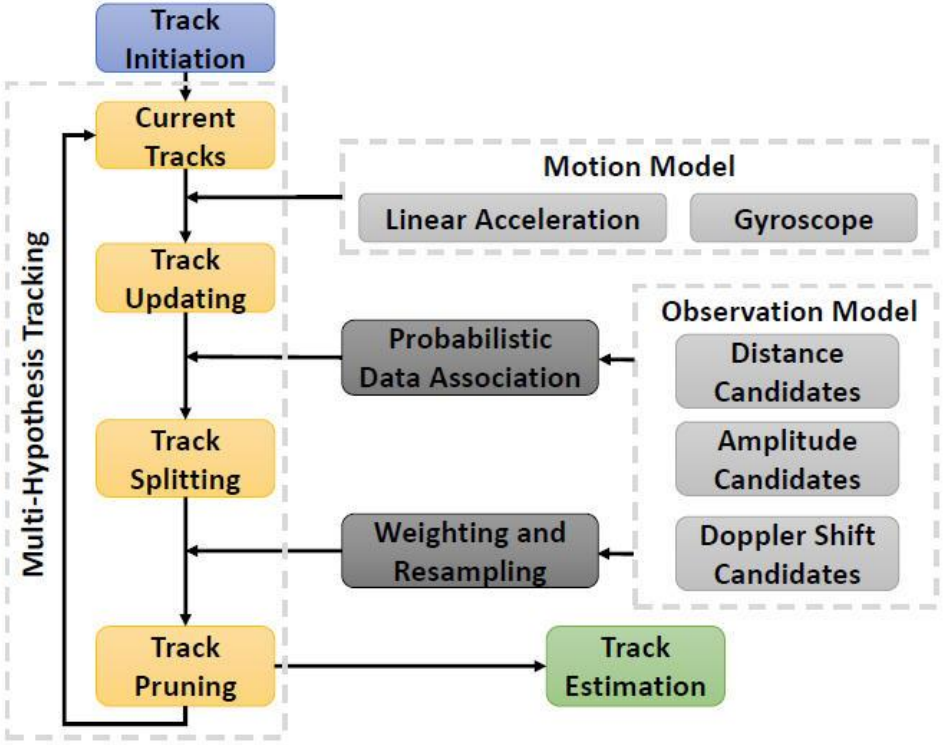


Figure 2: BatTracker uses inertial data in a motion model for track prediction, acoustics in an observation model for track correction, and fuse them into a multi-hypothesis tracking framework for robust tracking.

✓ 5.2.1.1 Motion Model

• Device position and velocity are defined as state vector $s(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]'$, where [x(t),y(t), z(t)] represents the device position in reference coordinate (i.e, room), $[\dot{x}(t), \dot{y}(t), \dot{z}(t)]$is the corresponding velocity along each axis.

• The absolute acceleration without gravity along each axis in the device's local coordinate is obtained from composite sensor linear acceleration from Android API, while the absolute device orientation is obtained from composite sensor game rotation vector.

• where ˆs(t + △t) is the predicted state at time t + △t , △t is the inertial data sampling interval, A(t) and B(t) are motion matrix from physical model $x(t) + \dot{x}(t)\triangle t + \frac{1}{2}a\triangle t^2 + \varepsilon(t)$, where a is the linear acceleration and ε(t) ∼ N(0, Σu) is the motion noise to capture the uncertainty.

• Since the acoustic sampling rate is lower than inertial, we keep updating the state during time interval [t , t + τ], where τ is the time interval between adjacent acoustic measurements.

$$\hat{s}(t + \triangle t) = A(t)s(t) + B(t)\mu(t) + \varepsilon(t) \qquad (2)$$

✓ 5.2.1.2 Observation Model

• The observation consists of range estimates to nearby objects at time t , which is represented as d(t) = $\{d_1(t), d_2(t), \ldots, d_N(t)\}$

• Note that a larger N can decrease the data missing probability, however it brings more clutter/noise measurements. As a balance between two cases, we set N = 15 in our experiments.

• There are actually two choices in our design: velocity derived from Doppler shift, and inferred by device position changes.

• Due to the limited sampling points for Doppler shift extraction, the velocity estimation is not robust enough as direct observations (evaluated in Section 6.1), Hence we select the later solution.

• It is assumed that distance measurements at each time t are normally distributed around the true ranges to objects with variance σ^2_r and a data missing probability P_M.

• In our case of 3 perpendicular surfaces in a corner, the 3 distances will uniquely determine the location of the device. Since the distance to each reference object can be the same, hence we can have N^3 possible combinations, which is a non-trivial amount.

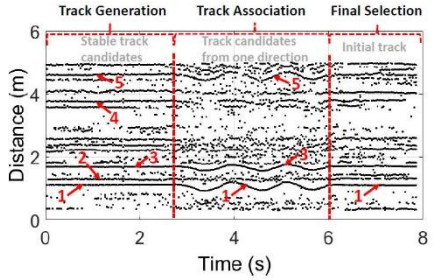


Figure 5: Track initiation process in one direction, which consists of track candidate generation, track association and final track selection.

✓ 5.2.2 Probabilistic Multi-Hypothesis Tracking

• The tracking problem is formulated as a Sequential Monte Carlo (SMC) problem, specifically, a variation of particle filtering framework with multi hypothesis tracking capability.

• We maintain a collection of K(t) “particle clouds", and each cloud represents a possible state estimate of a possible track, i.e, a track hypothesis.

• The framework operates on discrete time with an interval τ and repeats multiple steps for each time slot: track update, track splitting, track pruning and track estimation.

• Track update predicts the state s(t) of each particle in each existing particle clouds according to the motion model.

• Track splitting splits each particle cloud into “particle subclouds“ when multiple validated measurements are available, and computes the weight for each sub-cloud.

• Track pruning removes sub-clouds with a weight less than a certain threshold, normalizes the weights of remaining sub-clouds, then repopulating the number of particles within each sub-cloud to form a new set of particle clouds with a total number of particles of N.

• Track estimation returns the weighted mean of all the particles as the estimate of current state.

✓ 5.2.2 Probabilistic Multi-Hypothesis Tracking

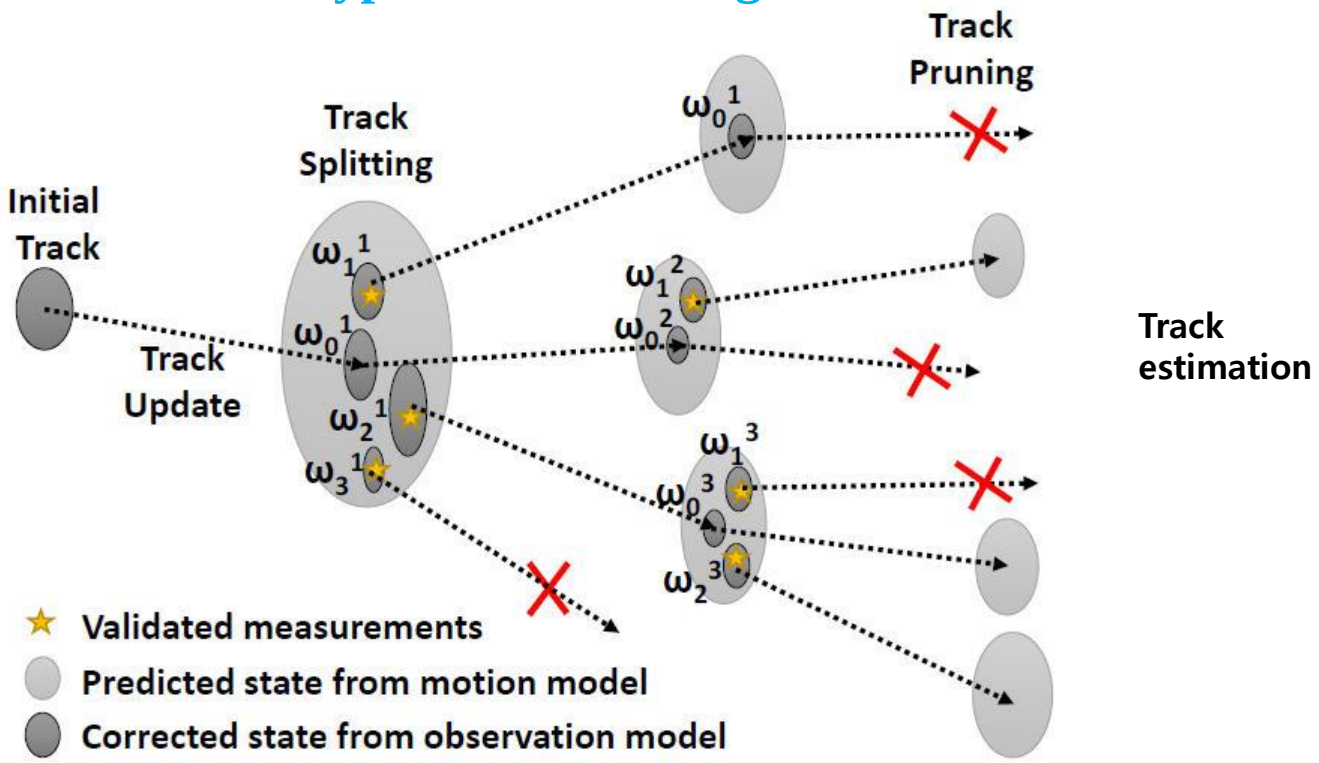


Figure 6: Tracks are represented by particle clouds, which are updated by motion model and split into multiple sub-clouds according to validated measurements. Those clouds with a weight lower than a threshold are pruned.

✓ 5.2.2.1 Track Update

• The state $\vec{s}(t) = \{s_1(t), s_2(t), \ldots, s_{K(t)}(t)\}$ consists of all the particle clouds s_k (t) indexed by k, where K(t) is the total number of particle clouds at time t . The kth particle cloud at time t is defined as:

$$s_k(t) = \{[x_k^n(t), y_k^n(t), z_k^n(t), \dot{x}_k^n(t), \dot{y}_k^n(t), \dot{z}_k^n(t)]\}_{n=1}^{N_k(t)} \qquad (4)$$

• where N_k (t) is the total number of particles in this cloud.

• For each iteration, the state represented by each particle is updated according to the state update equation (2) in motion model.

✓ 5.2.2.2 Track Splitting

• We leverage all the $M_k\ (t + \tau\ )$ validated measurements, and update a fraction of particles according to each measurement, hence splitting the particle cloud.

• A fraction of particles without update is always preserved to capture the case of data missing.

• When multiple validated measurements are available, we compute the likelihood for each of them that it is the “correct" measurement from our reference object using a probabilistic data association approach.

✓ 5.2.2.2.1 Probabilistic data association

• For each validated measurements in R_k (t +τ), we may have two situations: the “correct" measurement is captured, or missed.

• We denote the data missing probability as P_M(t), hence the weight for each measurement to be “correct" can be defined by a normalized weights:

$$\omega_m^k(t+\tau) = \begin{cases} \dfrac{\mathcal{L}_m^{*k}(t+\tau)}{P_M(t+\tau)+\sum_{j=1}^{M_k(t+\tau)} \mathcal{L}_j^{*k}(t+\tau)}, m = 1, 2, \ldots, M_k(t+\tau), \\ \dfrac{P_M(t+\tau)}{P_M(t+\tau)+\sum_{j=1}^{M_k(t+\tau)} \mathcal{L}_j^{*k}(t+\tau)}, m = 0, \end{cases} \tag{5}$$

$$\begin{aligned} \mathcal{L}_m^{*k}(t+\tau) &= p(z_m^k(t+\tau)|\hat{z}^k(t+\tau)) \\ &= \mathcal{N}[z_m^k(t); \hat{z}^k(t+\tau), S] \\ &= \frac{1}{\sqrt{2\pi S}} exp\{-\frac{1}{2}[z_m^k(t+\tau) - \hat{z}^k(t+\tau)]' S^{-1} \\ &[z_m^k(t+\tau) - \hat{z}^k(t+\tau)]\} \end{aligned} \tag{6}$$

✓ 5.2.2.2.2 Data association enhancement

• Now that we get an estimation of how likely the distance measurement combination $z^k_m(t+\tau)$ is the "correct" one based on the similarity between predicted position and distance measurement.

• we further optimize the data association by incorporating additional information from data: the velocity from Doppler shift and echo amplitude.

• From experiments, we found that θ has a strong impact of data missing from x-y plane, while φ has a strong impact on both x-z and y-z plane.

We define velocity $\upsilon^k_m(t+\tau)$ as the velocity measured from Doppler shift at time $t+\tau$, and $\alpha^k_m(t+\tau)$ is the vector of echo amplitudes.

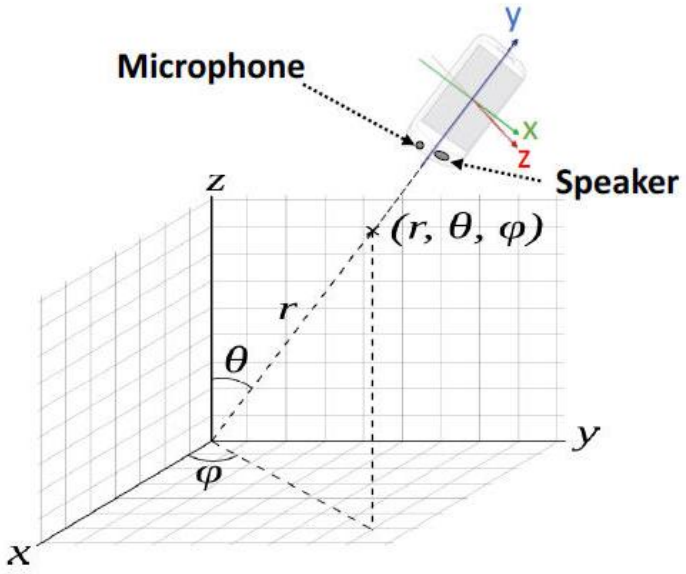


Figure 4: Microphone and speaker are at the bottom of the mobile device, which is pointed to a corner in a room for reliable echo detection.

$$\mathcal{L}^k_m(t+\tau) = \mathcal{L}^{*k}_m(t+\tau) \cdot p(\upsilon^k_m(t+\tau)|\hat{\upsilon}^k(t+\tau)) \cdot p(\alpha^k_m(t+\tau)|\alpha^k(t)) \quad (7)$$

$$P_M(t) = P_{MX}(\theta, t) P_{MY}(\varphi, t) P_{MZ}(\varphi, t) \quad (8)$$

✓ 5.2.2.2.3 Particle Cloud Splitting

• Each particle cloud may have multiple validated measurements, and their associated weights.

• We consider one validated measurement $z^k_m (t+\tau)$ at each time, hence our problem becomes a traditional particle filter problem, which requires weight calculation and resampling [43].

•Based on this measurement, we compute the weight for each particle within the kth cloud using the same form in Equation 7, and resample a number of $N^k_m (t+\tau)$ particles to form a sub-cloud, where $N^k_m (t+\tau)$ is proportional to $\omega^k_m (t+\tau)$ and $\sum_{m=0}^{M_k(t+\tau)} N_m^k(t+\tau) = N_k(t)$.

✓ 5.2.2.3 Track Pruning

• Each particle cloud is split into multiple sub-clouds at each time slot.

• we need to terminate some clouds and repopulate the remaining ones with new particles.

• it's split into 4 sub-clouds, each with a weight of ω_i , which is the data likelihood.

• We compare ω_i to a pruning threshold λ, and terminate clouds with weights lower than λ.

• Then we repeat the pruning, weight normalization, and repopulation steps.

• By setting a proper threshold, we keep the number of validated measurements within 10, typically $3-5$.

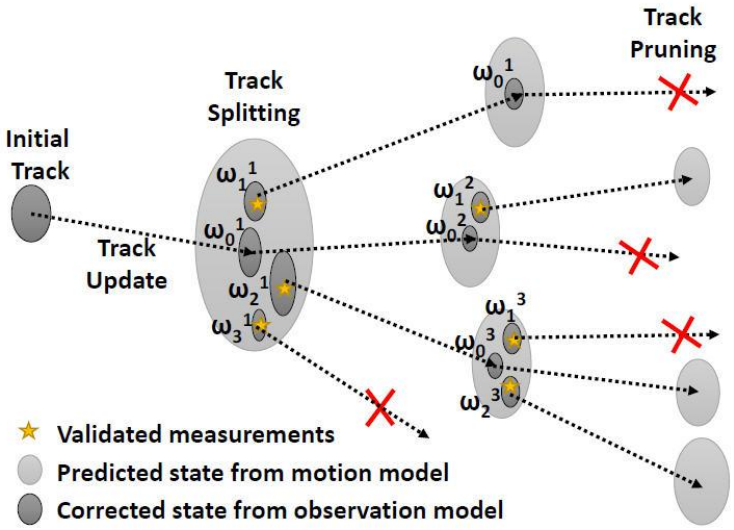


Figure 6: Tracks are represented by particle clouds, which are updated by motion model and split into multiple sub-clouds according to validated measurements. Those clouds with a weight lower than a threshold are pruned.

✓ 5.2.2.4 Track Estimation

• The estimated state of tracking is the mean of particle distributions and can be obtained as a weighted average of all the particle clouds:

$$s_E(t) = \Sigma_{i=0}^{K(t)} \omega_i s_i(t) \qquad (9)$$

• this estimated track is further smoothed using a moving average filter to reduce jitters.

• Our multi-hypothesis tracking algorithm repeatedly performs track update, track splitting, and track pruning over time, and generates track estimation for device tracking.

6. EVALUATION

KOREA UNIVERSITY

✓ Evaluation

• We use Huawei P9, a representative mobile phone of mainstream design with speaker and one microphone located at the bottom, as our mobile device to evaluate BatTracker from two aspects: acoustic measurements and tracking performance.

• We conduct tracking experiments in a highly cluttered laboratory with area of ~ 15m^2.

• Inertial data are sampled at 50Hz, and acoustic signal is sampled at 48KHz.

✓ 6.1 Acoustic Measurements

• Acoustic measurements consist of distance measurements to nearby objects and relative moving velocities from Doppler shift.

• We evaluate acoustic measurements using the following metrics:

• ranging accuracy, data missing probability, and velocity from Doppler shift.

✓ 6.1.1 Ranging Accuracy

• The ranging accuracy to reference objects is critical for accurate tracking.

• To evaluate the ranging performance, we select a plain wall in an empty space, and measure the distance to the wall at different locations using the mobile phone.

• We vary the location thus the ground truth distance changes from 0.5 − 3m with steps of 0.5m, and we repeat 30 times at each location.

• In quiet environment, the maximum error is within 2cm while the 90-percentile error is less than 1cm.

• Errors under background noise have slightly larger maximum error of ∼ 2.5cm, however, there's no significant performance deterioration.

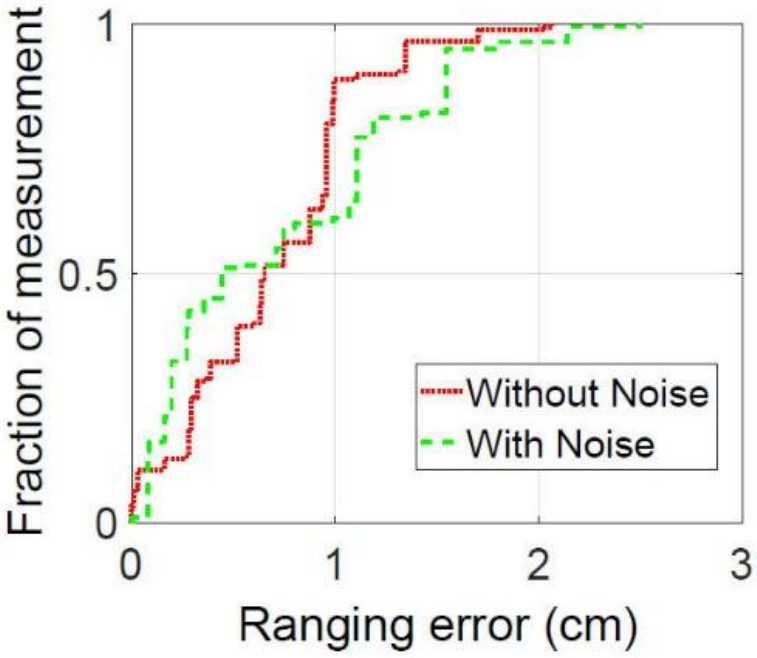


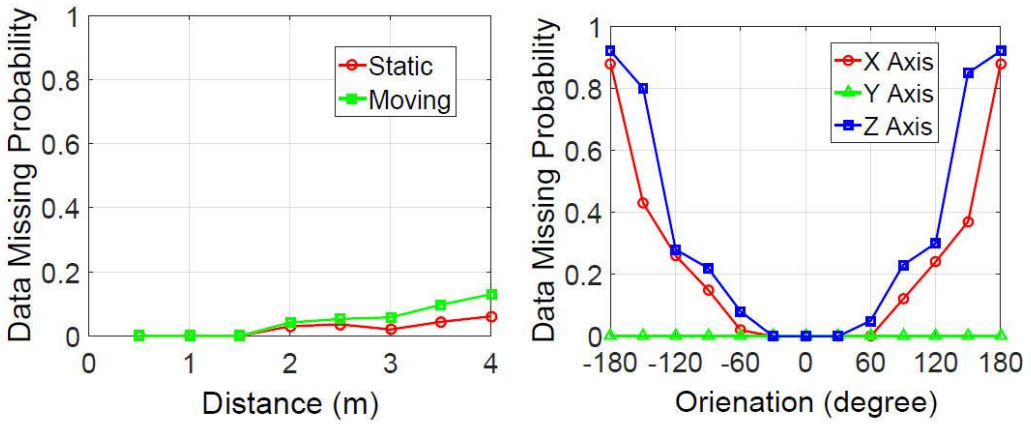
Figure 7: Distance candidate accuracy with and without background noise.

✓ 6.1.2 Data Missing Probability

• Data missing is one of the most challenging problems for robust tracking.

• From experiments, we find both the distance and relative orientation to an object have an impact on data missing probability, and phone movement can aggravate the problem.

• We evaluate the data missing probability in two individual experiments: impact of distance and movement, and impact of relative orientation.



(a) Distance impact on data missing. (b) Orientation impact on data missing.

Figure 8: Data missing probability under different distances and relative orientations.

✓ 6.1.2 Data Missing Probability

• We first evaluate the microphone sensitivity to sounds from different directions by analyzing the recording pulse amplitudes.

• One phone emits the designed signal continuously as sound source, simulating a mirrored speaker from a reference object.

• We use another phone to record the sound with a distance of 1.5m away from the sound source.

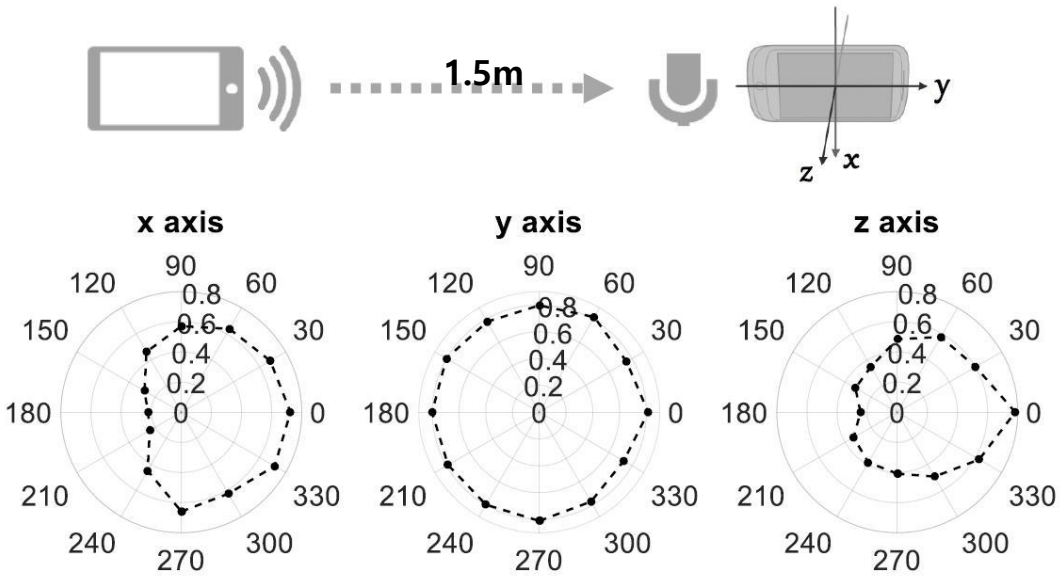


Figure 9: Normalized received signal amplitude under different rotations.

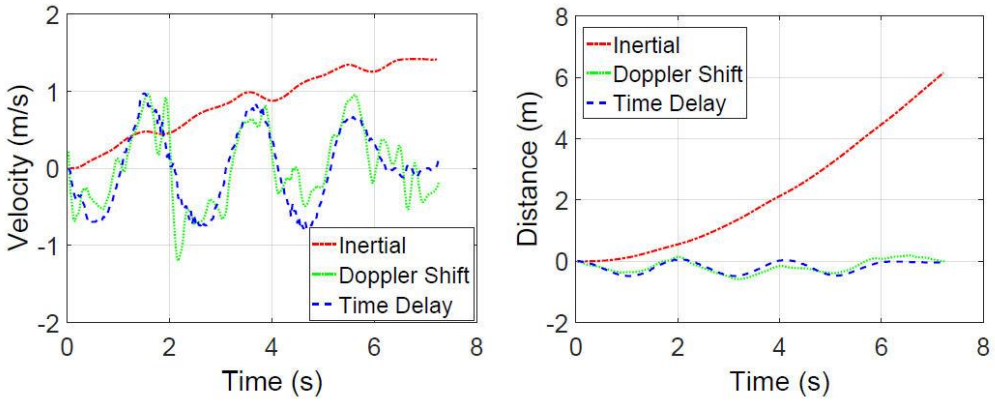
✓ 6.1.3 Velocity from Doppler Shift

• We evaluate the accuracy of velocity derived from Doppler shift, and compare the distance estimations with inertial sensor and sound propagation time delay.

• We choose a clean wall as reference object to simplify the data association with a distance of 1.5m between the wall and device.

• Then we move the phone back and forth, and compare the velocity and ranging from different schemes.

• when echoes from different directions overlap, we can not extract accurate frequency shift from the mixed signal.



(a) Device moving velocity. (b) Distance to the wall.

Figure 10: Comparison of velocity and ranging from different schemes: inertial sensor, Doppler shift, and sound propagation time delay.

✓ 6.2.1 Experiment Setup

• We evaluate the tracking performance from several aspects: tracking accuracy, various impact factors, comparison of different algorithms, and comparison with other work [19, 40].

• We evaluate BatTracker in a highly cluttered laboratory with large tables and cabinets (Figure 11(a)).

• We select the ceiling corner as potential reference objects.

• To quantify the tracking error, we use a tilted box with drawn traces, and move the mobile device along the traces.

• In such a way, we are able to get accurate ground truth for comparison.



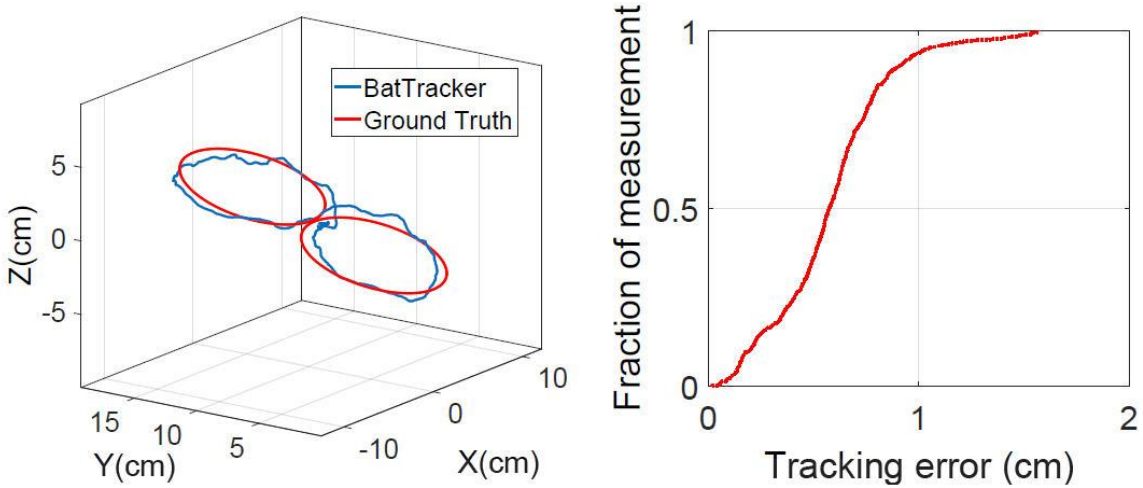
(a) Overview of experiment environment. (b) Tilted box with drawn traces.

Figure 11: We evaluate BatTracker in a highly cluttered laboratory, and a tilted box with drawn traces used as ground truth.

✓ 6.2.2 Tracking Accuracy

•First, we evaluate the 3D tracking accuracy of BatTracker (2D tracking accuracy is evaluated later in the comparison with other work).

• Figure 12(b) shows the CDF of 3D tracking error, which is less than 1cm at 90-percentile, and the maximum error is ~ 1.5cm.



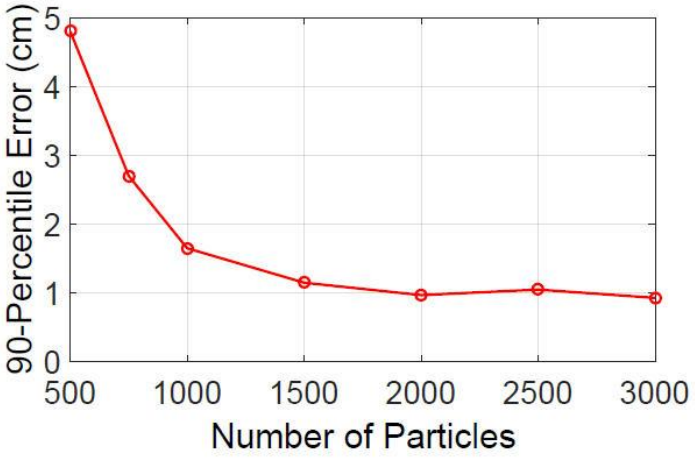
(a) Tracking result and ground truth. (b) Tracking error.

Figure 12: 3D tracking result and CDF.

✓ 6.2.3 Impact of Number of particles / Background Noise

• Figure 13 shows that the 90-percentile error keeps decreasing from ~ 5cm to ~ 1cm when the number of particles increases from 500 to 1500, then it stays relatively stable.

• Our designed signal is at 17KHz, which is far from common sound noises (usually from 1−8KHz), hence we can easily filter out low frequency noises.



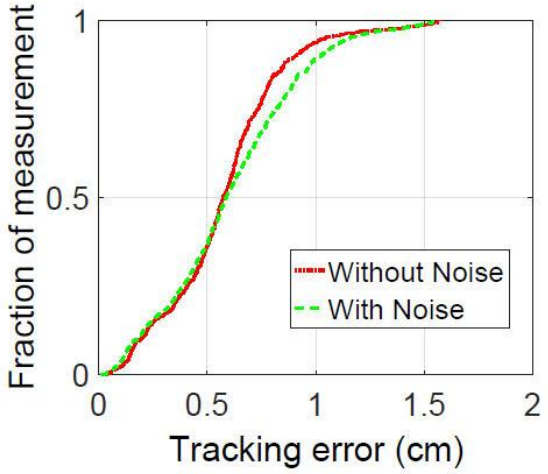


Figure 13: Tracking error decreases as the number of particles increases, then becomes stable.

Figure 14: Impact of background noise.

✓ 6.2.4 Resistance to Track Initiation Error

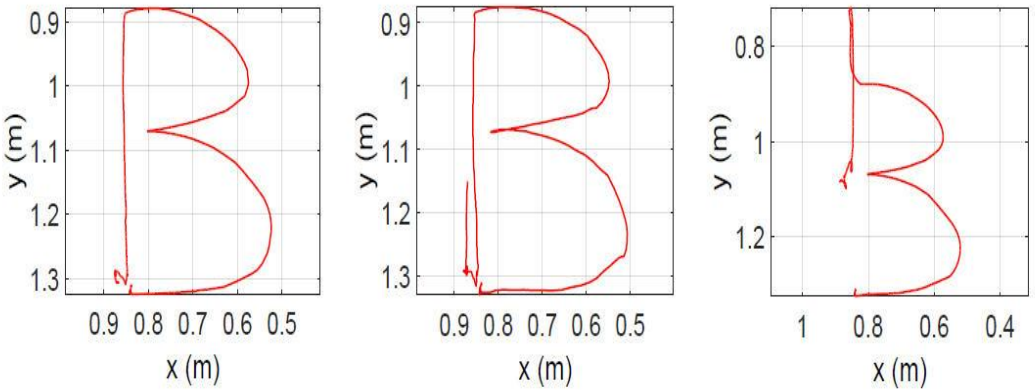
• To initialize a track, the user needs to move the phone in three directions back and forth sequentially to get the distances to three reference objects.

• These minute movements may introduce initiation error since the device can not be guaranteed to return back to the same location.

• To evaluate the impact of initiation error on the tracking accuracy, we manually add initiation error on the initial track.

• Figure 15(b) shows the result by manually adding an error of 15cm in y axis.

• We observe that the initiation error is corrected near the beginning, and the overall tracking result is not obviously impacted.



(a) Accurate initiation. (b) Medium initiation error. (c) Large initiation error.

Figure 15: Tracking results with different level initiation error.

✓ 6.2.5 Comparison with Different Algorithms

• We compare the performance of different algorithms, which includes naive nearest neighbor range-only tracking, inertial only tracking, single hypothesis conventional particle filter tracking, and our multi-hypothesis tracking.

• We use the same data set ("ACM" written in the air), and apply the above algorithms for comparison.

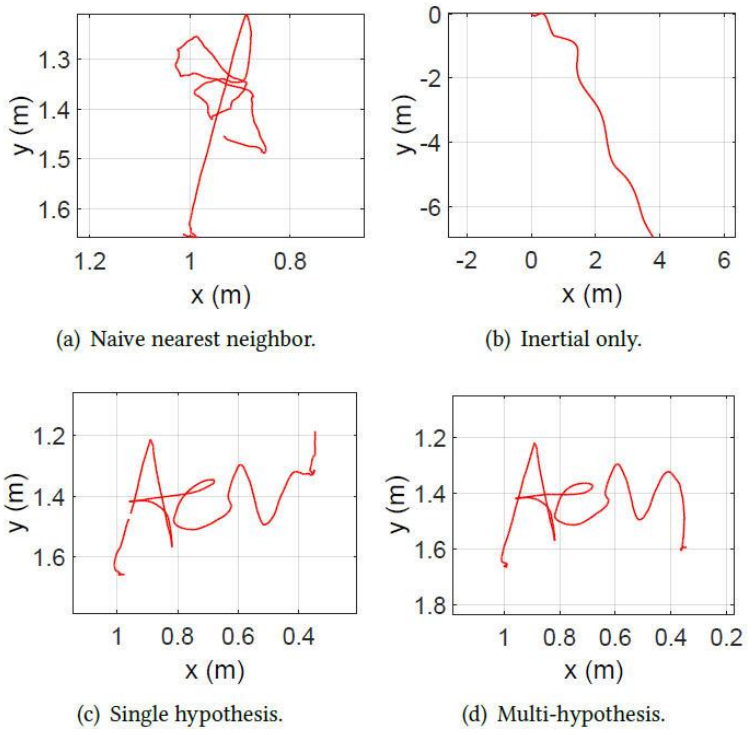


Figure 16: Tracking results of different approaches. Naive nearest neighbor tracking is easily lost, while the inertial only tracking drifts rapidly. Single hypothesis particle tracking is more robust but still lost due to track deviation. BatTracker's multi-hypothesis tracking outperforms others.

✓ 6.3 Drawing Evaluation

• In this comparison, BatTracker requires clean walls as reference objects, while CAT/AAMouse require external speakers.

• CAT develops a FMCW(frequency-modulated continuous wave) based approach which can accurately measure the distance from mobile device to external speakers.

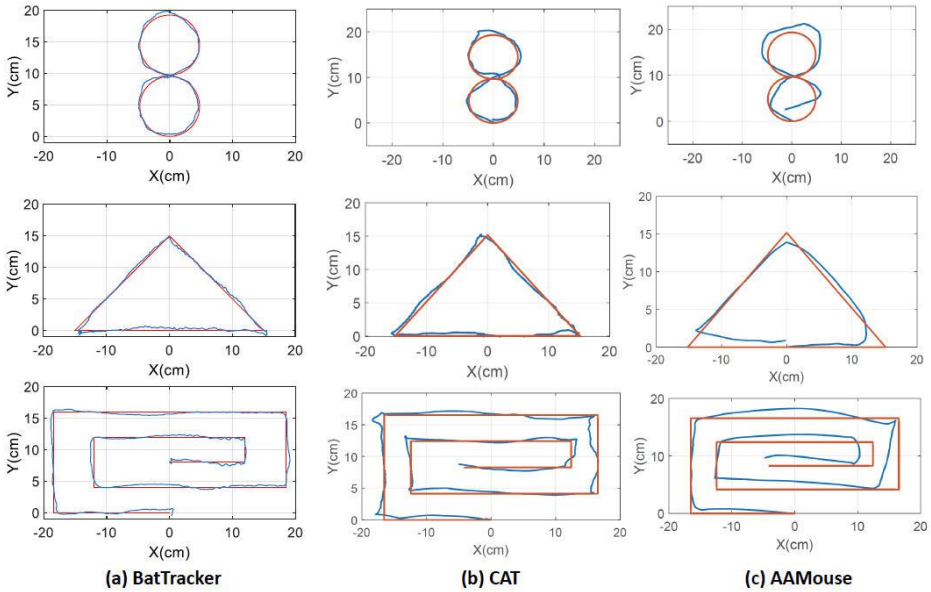
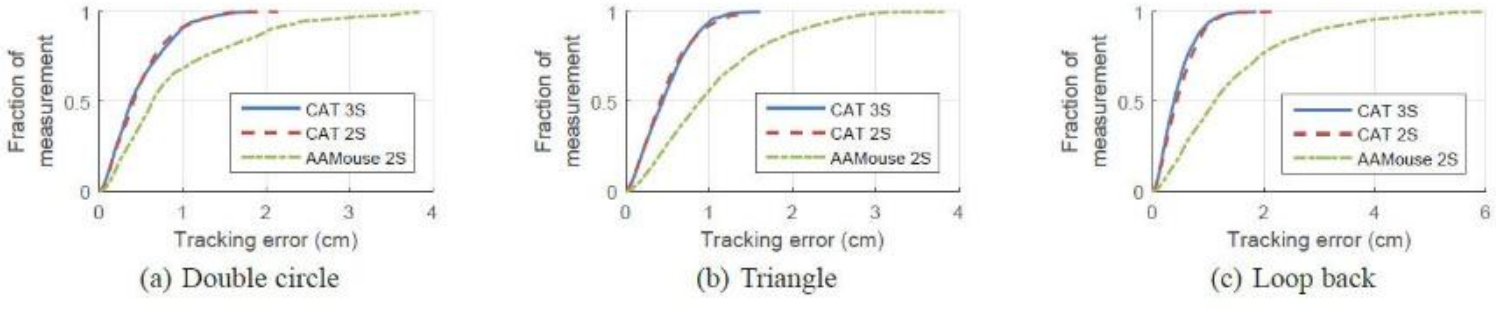


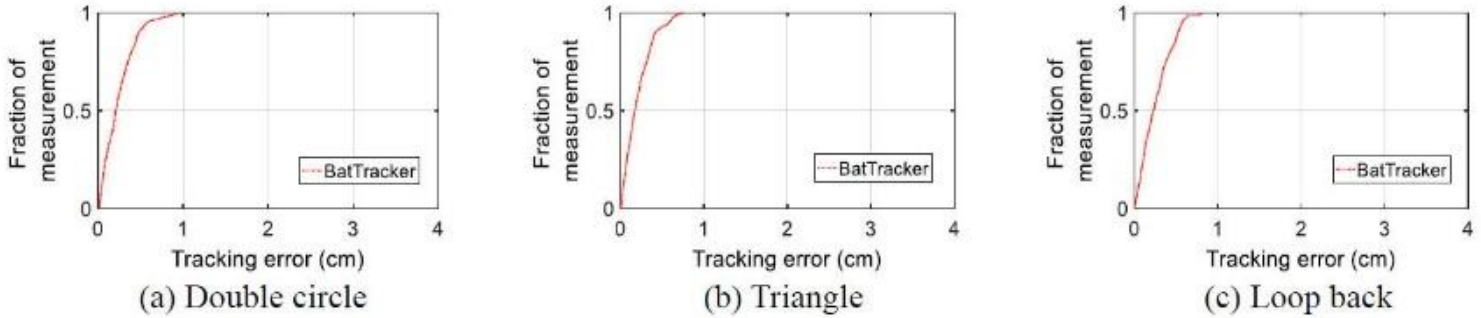
Figure 17: Shapes created by BatTracker, CAT, and AA-Mouse.

✓ 6.3 Drawing Evaluation

• CAT measures the distances from device to anchor speakers, and triangulate the location of the mobile device. Such triangulation can enlarge the tracking error.



(a) Drawing error of CAT and AAMouse.



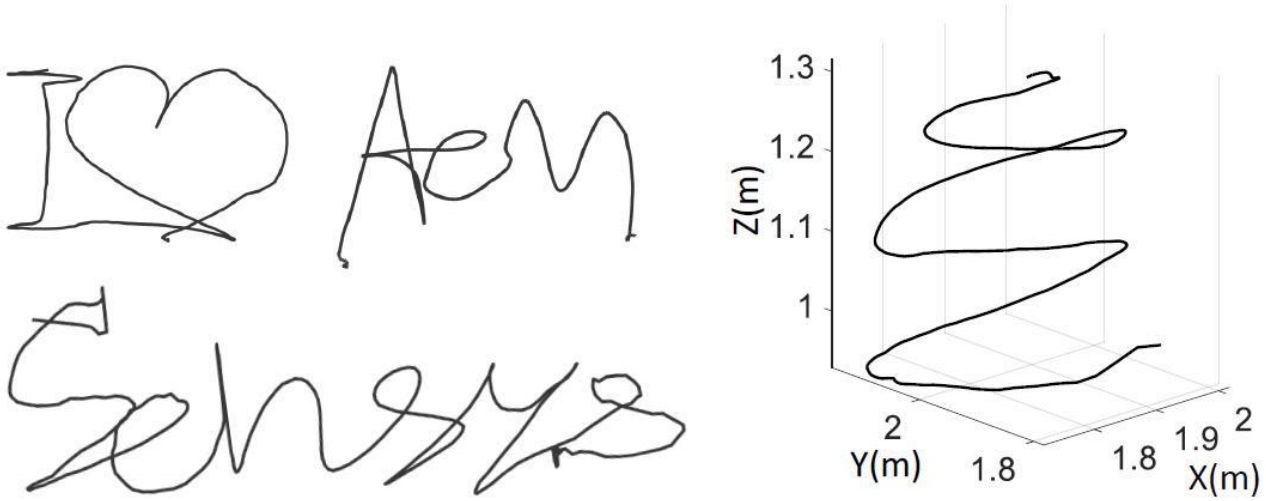
(b) Drawing error of BatTracker.

Figure 18: The drawing error comparison between different schemes. The results in figure (a) are from the CAT paper [19], and two figures (a, b) are drawn separately because we do not have the raw data in figure (a).

✓ 6.3 Drawing Evaluation

• Note that there is a sharp jump at the beginning at the first letter “S" in the word “Sensys“. This is caused by the initiation error, and corrected automatically in a very short period.

•The spiral shows BatTracker is able to produce 3D tracking with free movements.



(a) “ I ♡ ACM Sensys". (b) Spiral.

Figure 19: “ I ♡ ACM Sensys" and a spiral drawn in the air.

KOREA UNIVERSITY

✓ 6.4 Computation Complexity

• We implement the core tracking algorithms on Android devices, and develop a real-time tracking application.

• All the computation are done in real-time on the smart phone, while the tracking results are streamed to laptop through Wi-Fi for display only.

• Further increasing particles to 3000 causes serious lag, while the CPU usage is still under 30%, which indicates the computation potential is not fully exploited.

• There's still a large room for optimization, such as a better multi-threading design.

Table 1: Allocated memory and CPU usage with different numbers of particles.

# particles	600	900	1200	1500	1800
Memory (*MB*)	9.56	11.52	12	12.33	13.02
CPU (screen on)	16.5%	17.74%	18%	18.36%	19.65%
CPU (screen off)	9.6%	11%	11.7%	12.01%	13.74%

7. DISCUSSION

✓ 7.1 Potential Use

• High precision, infrastructure-free device tracking in indoor environments has broad application scenarios.

• The latest video games and VR devices rely on additional anchor devices for high precision tracking, and gesture tracking requires inertial sensors on wearable devices which are always inaccurate.

• To the best of our knowledge, BatTracker is the first to achieve high precision tracking without any additional hardware, which makes it very convenient and attractive for indoor device tracking.

✓ 7.2 Limitations

• Despite the high accuracy and infrastructure-free convenience we have, there exist multiple limitations which need to be addressed in future work.

(1) Tracking range. We limit the range of reliable tracking in our current design within a space of $3*3*3m^3$.

(2) Device holding gesture. Due to the hardware limitations, users have limited range of device rotation to ensure reliable acoustic measurements.

(3) Reference Objects. For optimal performance, BatTracker uses three perpendicular plains as reference objects, which could be the ceiling and side walls in a room, or large furniture such as closets, cabinets, and tables.

(4) Track loss problem. Despite the sophisticated algorithms that we propose for robust tracking, BatTracker still has a small chance of track loss, especially when the device is not held properly or is too far away from reference objects.

✓ 7.3 Future Work

• Our future work for BatTracker focuses on improving tracking robustness, especially dealing with track loss problem.

(1) Fast track recovery. We plan to design a mechanism for automatical track loss detaction and recovery. One intuition is from the match between inertial data and acoustic measurements.

(2) Utilize all the available objects. Our current design only utilizes three large objects as reference, hence a large portion of measurement information from other objects are not used.

(3) Customized hardware. Adding customized omni directional, high-sensitivity microphones or multiple orthogonal microphones into existing devices can enhance the tracking robustness.

(4) Variations among different smart phones. To make BatTracker ubiquitous, more comprehensive tests on different smart phones are needed as our future work.

8. CONCLUSION

• we propose BatTracker, which incorporates inertial and acoustic data for robust, high precision and infrastructure-free tracking in indoor environments.

• BatTracker leverages echoes from nearby objects instead of external infrastructure, thus requires less cost, deployment efforts, and is more convenient to use.

• A probabilistic multi-hypothesis tracking algorithm creates, prunes and evolves multiple track hypotheses based on measurement evidences to accommodate uncertainty in device position.

• Experiments in real environments show that BatTracker can track a mobile device's movement in 3D space at sub-cm accuracy, comparable to the state of-the-art infrastructure based approaches, while eliminating the needs of any additional hardware.

9.CRITIQUE

KOREA UNIVERSITY

• 음향의 벽 반사 정보를 통해 움직임을 측정하는 방법 자체가 매우 획기적인 연구로 판단됨.

• 여러 제약사항은 있지만, 음향을 이용해 1~2cm 이내의 오차 정확도로 움직임을 추적할 수 있다는 결과가 놀라움.

• 역으로 음향을 통해 SLAM(Simultaneous localization and mapping) 기술로 응용할 수 있을 것으로 보임

Q & A
감사합니다