Closing the Gaps in Inertial Motion Tracking

Sheng Shen
University of Illinois at Urbana-Champaign
sshen19@illinois.edu

Mahanth Gowda
Pennsylvania State University
mahanth.gowda@psu.edu

Romit Roy Choudhury
University of Illinois at Urbana-Champaign
croy@illinois.edu

ABSTRACT

A rich body of work has focused on motion tracking techniques using inertial sensors, namely accelerometers, gyroscopes, and magnetometers. Applications of these techniques are in indoor localization, gesture recognition, inventory tracking, vehicular motion, and many others. This paper identifies room for improvement over today's motion tracking techniques. The core observation is that conventional systems have trusted *gravity* more than the *magnetic North* to infer the 3D orientation of the object. We find that the reverse is more effective, especially when the object is in continuous fast motion. We leverage this opportunity to design *MUSE*, a magnetometer-centric sensor fusion algorithm for orientation tracking. Moreover, when the object's motion is somewhat restricted (e.g., human-arm motion restricted by elbow and shoulder joints), we find new methods of sensor fusion to fully leverage the restrictions. Real experiments across a wide range of uncontrolled scenarios show consistent improvement in orientation and location accuracy, without requiring any training or machine learning. We believe this is an important progress in the otherwise mature field of IMU-based motion tracking.

CCS CONCEPTS

• **Human-centered computing** → *Mobile computing*; *Ubiquitous and mobile devices*; • **Hardware** → *Digital signal processing*;

KEYWORDS

IMU, Motion Tracking, Dead Reckoning, Sensor Fusion, Orientation, Location, Magnetometer, Accelerometer, Gyroscope

MobiCom '18, October 29-November 2, 2018, New Delhi, India

ACM ISBN 978-1-4503-5903-0/18/10…$15.00
https://doi.org/10.1145/3241539.3241582

ACM Reference Format:
Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. 2018. Closing the Gaps in Inertial Motion Tracking. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3241539.3241582

1 INTRODUCTION

Inertial sensors (IMUs) serve as the bedrock to a large number of mobile systems and applications. Smartphones and smartwatches have already utilized IMUs to infer human activities and gestures, while drones and robots have classically employed IMUs to guide their motion-stabilization and control algorithms. More recently, IMUs are playing a role in everyday objects. A start-up called Grush [1] proposed a fascinating idea that won the "2016 America's Greatest Makers" contest. The company tracks the motion of *IMU-embedded toothbrushes* and feeds this motion into a smartphone game where monsters need to be killed. When brushing teeth, a child must move to different corners of his mouth to kill the scattered monsters in the smartphone screen. In a more serious context, health rehabilitation centers are increasingly giving motion-trackers to patients so their progress can be monitored even at home. Needless to say, any improvement to IMU-based motion tracking will impact a range of systems and applications.

Let us begin by intuitively understanding the core problems in IMU-based motion tracking. Observe that all the inertial sensors, i.e., accelerometer, gyroscope, and magnetometer, operate in their local frames of reference. For instance, if the accelerometer measures motion along its X axis, it is not clear what this motion means in the global reference frame. As an analogy, imagine a friend calling from inside a flying airplane and saying that she is turning "right". Without knowing the 3D orientation of her plane in the earth's reference frame, there is no way to infer which way she is turning. Put differently, tracking the motion of any object first requires the knowledge of the object's 3D orientation in the *global framework*. Then, the motion sensed locally by the IMU sensors can be appropriately projected onto the global framework, ultimately enabling a meaningful motion tracking solution.

Precisely estimating the object's 3D orientation in the global reference frame (GRF) is non-trivial. Conventional systems solve this by utilizing gravity and magnetic North as "anchor" directions. Loosely, the accelerometer measures components of gravity along its 3 axes and infers how tilted the object is from the horizontal plane. Once this tilt is compensated, the magnetometer can measure the object's heading angle by comparing against the magnetic North direction. This indeed yields the 3D orientation of the object, *although only in the case when the object is completely static*. When the object is moving, the accelerometer measures the "*mixture*" of both gravity and linear motion, making gravity isolation difficult. As a consequence, the object can no longer be tilted precisely to become flat on the horizontal plane, which further pollutes the estimation of the magnetic North direction. In summary, estimating the global 3D orientation of a moving object is the key bottleneck to IMU based motion tracking. Worse, this 3D orientation needs to be continuously tracked, since local sensor measurements need to be continuously projected onto these global directions.

Of course, we are not the first to look into this problem. Many techniques exist in literature [2–10], of which A^3 [11] from MobiCom 2014 is probably the most effective. Importantly, A^3 actually sidesteps the problem of decomposing gravity and linear motion from their mixture; instead, it opportunistically searches for moments when the object is static. The object's 3D orientation is re-calibrated at these static moments (using gravity and North), and for all times between these moments, A^3 uses the gyroscope to track the local changes in orientation. Referring back to our analogy, if our friend from the airplane could periodically stop and tell us her global orientation, we could utilize her local measurements of "turning right" to interpolate her global orientations at all times.

Unfortunately, objects may not pause often, and even if they do, determining those moments produces false positives and false negatives. All in all, A^3's performance upholds only in certain types of movements, and even then, some components of IMU information remain unused (as elaborated later).

This paper, inspired by A^3, finds room for improvement. Our core insight is that the *earth's 3D magnetic North vector* could serve as a better global "anchor" than gravity, especially during motion. This is because magnetometers are mostly unpolluted by the device's motion, and as a result, can always measure the magnitude of the global 3D North vector. The direction is still unknown since it is a function of the object's own orientation. Nonetheless, if we can measure the direction of this North vector just one time, we can utilize it thereafter as a trusted anchor for tracking orientation, even when the object is moving. Gravity is still necessary, but only as a secondary anchor to complete the orientation estimation.

This sensor fusion opportunity ultimately results in *MUSE*, an iterative algorithm that uses gravity from a static moment to estimate the global 3D North vector, and thereafter, uses this North vector as the primary anchor. Gravity is still used, although with varying trust, depending on whether the object is moving. Finally, the gyroscope measurements are also used to track rotation, and then fused with the accelerometer + magnetometer measurements to overdetermine the system. The net result is improved 3D orientation in the global reference frame (GRF), in turn helping the localization aspect of motion tracking.

Finally, when the object's motion is restricted (i.e., it does not span all the 6 degrees of freedom), the mixture of gravity and linear motion can be better decomposed from the accelerometer. This is because the space of possible movement is now restricted, allowing for better guesses on how the mixture may be formed. We exploit this opportunity as well by jointly estimating orientation and location from the same accelerometer data (as opposed to determining orientation first and then inferring linear motion). To this end, we feed all sources of IMU information into a Bayesian framework – an augmented particle filter – that will ultimately overdetermine the system, and output both 3D orientation and location in the global reference frame.

We implement and evaluate *MUSE* on two off-the-shelf platforms: (1) a Samsung Galaxy S6 smartphone, and (2) a Samsung Gear Live smartwatch. Orientation tracking is evaluated across various human activities and object motion. Ground truth is obtained by periodically bringing the device to a pre-specified orientation. Comparison with A^3 shows an average of 2.9X performance gain for various natural activities, and higher when the motion is continuous without pauses. We also ask humans wearing the smartwatch to move their hands normally, and jointly track the watch's orientation and location in an online manner. Ground truth from Microsoft Kinect reveals a median location error of 8.8cm, better than the state-of-the-art method that requires full future data for offline decoding.

We summarize *MUSE*'s contributions in 2 parts:

- Part I identifies that the *global 3D magnetic North vector can serve as a better anchor compared to gravity*, and utilizes it for tracking the 3D orientation of a moving object.
- When motions are restricted, Part II designs a particle filter based technique that *jointly estimates location and orientation, as opposed to estimating them serially* (i.e., orientation first, and then location).

The rest of this paper elaborates on each of these contributions, however, we need to begin with the foundations of IMU based motion tracking. While this makes the next section long, we believe the material is necessary to appreciate the problems and solutions. We also believe the material is easy to follow, since it starts from first principles.

2 FOUNDATIONS OF TRACKING

■ **Reference Frames:** Consider the general case where an object's motion, i.e., sequence of 3D locations and 3D orientations, needs to be tracked in a global reference frame, say ⟨*North*, *East*, and *Up*⟩. The sequence of 3D locations, when differentiated twice, gives acceleration; the accelerometer measures this acceleration, but in its local reference frame (LRF). The sequence of 3D orientations, when differentiated once, gives angular velocity; the gyroscope measures this angular velocity, but again in the LRF. Tracking 3D motion in the global reference frame (GRF) requires a continuous translation between the two coordinate frames. Specifically, at every time instant, the object's LRF needs to be rotated and aligned to the GRF, and the acceleration needs to be computed in this aligned framework. Thus, the first question in tracking degenerates to constantly estimating this LRF-to-GRF rotation.

■ **Understanding 3D Orientation:** The LRF-to-GRF rotation is essentially the phone's 3D orientation. To understand, consider a plane taking off in Figure 1, with its true 3D heading direction as 45° North-East and 30° to the vertical direction. However, for a passenger inside the plane, the heading direction is always along its local Y axis. Thus, computing the global heading direction entails rotating the plane −30° around its local X axis (which is along the wing of the plane), and then rotating the plane −45° around its local Z axis. These rotations align the local and global axes, capturing the orientation mismatch between the LRF and GRF. Mathematically, this net mismatch can be modeled as a single 3D rotation matrix, and the inverse of this matrix (i.e., the mismatch from GRF to LRF) is defined as the object's 3D orientation.

■ **Computing 3D Orientation:** In reality, how would a passenger inside the plane compute its global heading direction? One opportunity is to measure quantities, such as gravity and magnetic North, whose directions are universally known in the global reference frame (GRF). Specifically, an accelerometer would be able to identify that the universal gravity vector is tilted by 30° from its local $-Z$ axis. Similarly, a magnetometer (or compass) should be able to recognize that the Earth's magnetic North direction is offset by 45° from its local Y axis. Thus, using local measurements of gravity and North from the accelerometer and the magnetometer

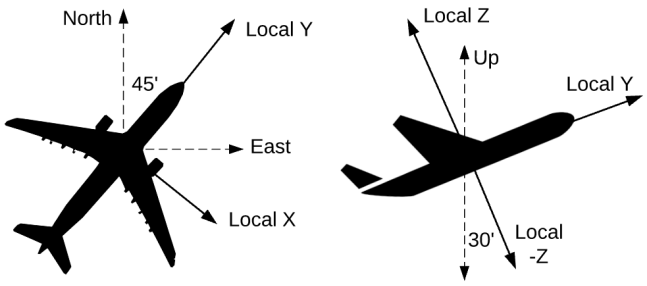

Figure 1: The top view (left) and side view (right) of a plane taking off. The 3D orientation of the plane is the net rotation needed to align the plane's local ⟨*X, Y, Z*⟩ axes with the global ⟨*North, East, Up*⟩ axes.

respectively, an object should be able to compute its own LRF-to-GRF mismatch, which is exactly the 3D orientation of the object[1].

■ **Basic Motion Tracking:** Now, tracking the object's *3D orientation* over time can be achieved in two ways: (1) performing the above LRF-to-GRF alignment (using gravity and North) at every instant of time, or (2) performing the alignment once to get initial orientation, and then integrating the gyroscope data thereafter to obtain subsequent orientations. Tracking the object's *3D location* over time is slightly more involved. For each time step, the object's 3D orientation needs to be estimated and the accelerometer data needs to be projected to this global reference frame. This projected accelerometer data contains both linear acceleration and a gravitational component (Figure 2(a)). After removing gravity, linear acceleration is now double integrated to compute the next location. The orientation and location estimation process together repeats for every time step, ultimately producing the 3D orientation and 3D location of the object at any given time.

2.1 Why Theoretical Tracking Falls Short

The above tracking method, although conceptually complete, does not scale to real world situations. We discuss 3 main issues:

(1) Gravity Pollution: In describing how the accelerometer uses gravity to compute its vertical misalignment, we need to assume that the object is static. Otherwise, the object's motion will mix with gravity measurements, yielding an incorrect tilt. Put differently, computing the vertical tilt of a moving object is difficult.

[1]Note that only gravity or only North is inadequate for determining 3D orientation. For instance, even if gravity is perfectly aligned along the $-Z$ axis, the object can still be in many possible orientations on the horizontal plane (with different heading directions).

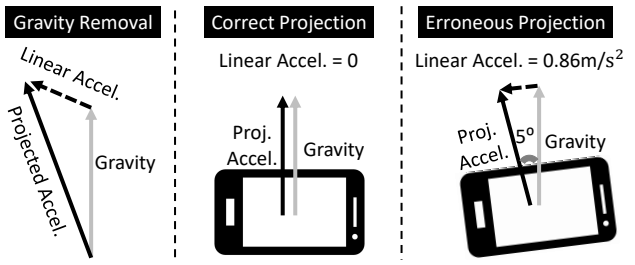

Figure 2: (a) Accelerometer projected to GRF contains both linear acceleration and gravity. (b) Take a static object as an example, projection with correct orientation removes gravity perfectly, making no errors in linear acceleration, but (c) slight offset in orientation can cause large projection error, leading to wrong linear acceleration.

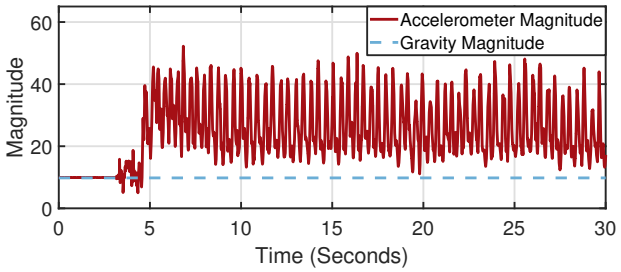

Figure 3: The magnitude of accelerometer measurement vs. the constant magnitude of gravity, as a user picks up a phone in hand and starts running. Clearly, the accelerometer cannot measure gravity properly in this case.

(2) Magnetic Interference: In indoor environments, magnetometer measurements of the earth's magnetic North can be polluted by nearby ferromagnetic materials. This again derails the estimate of 3D orientation, impacting 3D location as well.

(3) Inherent Sensor Noise: Finally, hardware noise is inherent in all IMU sensors. Any integration operation accumulates this noise and the problem is pronounced for location tracking with accelerometers. This is because accelerometers need to be integrated *twice* to obtain location, and further, any orientation error directly translates to accumulated location-error over time. Gyroscopes also drift but relatively less since they require a single integration. Magnetometers do not drift but experience a random high frequency noise in their measurements.

We emphasize again the perils of 3D orientation error on location. Observe that the accelerometer data would get projected erroneously from LRF to GRF (Figure 2(b) and 2(c)), and as subsequent velocities and locations get computed in GRF (using single and double integrations), the error will diverge over time. The analogy is in orienting a gun slightly off from the direction of the target – the margin by which the bullet misses the target increases with the distance of the target from the gun. Thus, precisely estimating 3D orientation is crucial and challenging, especially for a moving object in an indoor (ferromagnetic) environment.

With this background on practical challenges, we zoom into today's techniques and distill the room for improvement.

2.2 State-of-the-Art Method

Classical motion tracking spans control theory, robotics, signal processing, graphics [9, 10], and is difficult to cover in the interest of space. The recent work called A^3 [11] from MobiCom 2014 is perhaps the most practical solution today. As mentioned earlier, A^3 recognizes that accelerometers measure the mixture of gravity and motion; that isolating gravity is difficult; and that 3D orientation is difficult to estimate. In light of this, their proposal is to identify opportunities when gravity measurement is unpolluted. This happens in 2 cases: (1) when the object is static, or (2) when the object is in pure rotational motion (i.e., only rotating but not moving). In both these cases, the accelerometer only measures gravity, allowing for estimating the global orientation.

The static case is easy to detect, but for pure rotational motion, A^3 showed that the gyroscope measurement correlates well with the accelerometer data, since gravity will spin similarly in the object's local reference frame. Falling back to the plane analogy, consider Alice sitting inside a plane that is not moving but only spinning in the air (and assume Alice has a gyroscope in her hand). Also assume that Alice can track how the direction of the sun is changing (perhaps because the plane is made of transparent glass). A^3 points out that Alice would see the sun spinning around her, and this spin should correlate with the spin measured by her gyroscope. However, if the plane was both spinning and moving linearly, the correlation would break down. Thus, strong correlation is an indication of unpolluted gravity, offering an opportunity to align LRF to GRF, and ultimately infer the device's 3D orientation.

We believe that A^3 is an elegant contribution, however, the shortcoming is that such opportunities are infrequent. Figure 3 shows a case of running, where the accelerometer magnitude is constantly varying. Given running activities can easily last for far longer time durations, A^3 may not be able to utilize the pausing opportunity at all. For many real-world movements, the state of "rotation but no acceleration" also occurs rarely – for the running case in Figure 3, as well as many other evaluated later, we did not find a single opportunity. As a result, 3D orientation tracking still remains

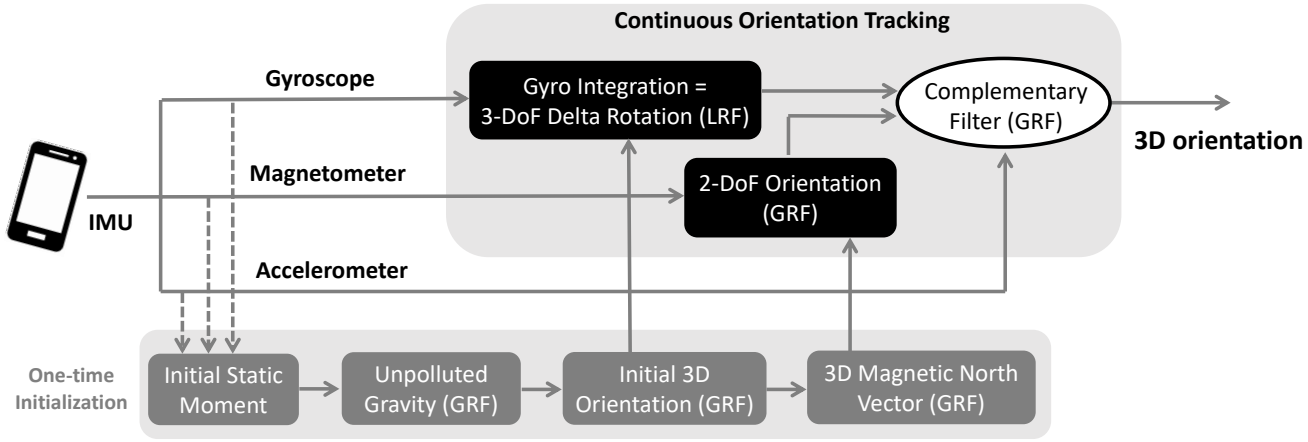

Figure 4: *MUSE* processing pipeline: The IMU data is processed in stages to compute the 3D orientation in the global reference frame. The gyroscope is integrated to provide 3-DoF information on orientation, while the magnetometer produces extra information on 2 DoFs. These two sources of information are fused in a complementary filter. The accelerometer is opportunistically used to refine 3D orientation. A one-time initialization step is necessary to bootstrap the system, during which the initial orientation and the 3D magnetic North vector are computed.

an elusive problem, hindering accurate location tracking. In light of this, we present our proposal on magnetometer fusion, *MUSE*.

3 PART I: ORIENTATION ESTIMATION

This section designs improvements to 3D orientation estimation. Note that A^3 and prior methods have always viewed gravity as the primary "anchor" for estimating orientation, and since gravity gets mixed with linear motion (in the accelerometer), it is difficult to extract a precise global orientation. We break away from this approach and observe that 3D magnetic North vector can be a more effective anchor. The advantage arises from magnetometers being unaffected by linear motion of the object. However, the tradeoff is that the intensity and direction of magnetic North may vary across locations (unlike gravity). In light of this, *MUSE* requires the object to start from a static moment, utilizes the unpolluted gravity to precisely estimate the 3D North vector, and thereafter uses the North vector as the anchor for orientation. We elaborate the algorithm next.

3.1 MUSE Overview

Figure 4 shows the *MUSE* orientation estimation pipeline. IMU data (accelerometer) from the initial static time window offers unpolluted gravity, used to determine the vertical tilt of the object. The magnetometer, on the other hand, measures the 3D magnetic vector in its local reference frame (LRF), and projects it to the horizontal plane to compute the object's heading. These two anchors together fully determine the 3D orientation of the object in the global reference frame (GRF)[2]. Now, once this initial 3D orientation $\boldsymbol{O}(t_0)$ is known, the current magnetometer's local measurement, $\boldsymbol{N}^L(t_0)$, can be projected back onto the GRF, leading to the global 3D magnetic North vector, $\boldsymbol{N}^G$. This gives us the anchor we need:

$$\boldsymbol{N}^G = \boldsymbol{O}(t_0)\boldsymbol{N}^L(t_0) \quad (1)$$

As the object starts moving, the magnetometer tracks $\boldsymbol{N}^G$ in its LRF, leading to 2 DoFs (degrees of freedom) of global orientation. In a parallel thread, gyroscope tracks all 3 DoFs of rotation. Together, this is an overdetermined system, with 5 DoFs of information available for 3 DoFs of changing orientation. To avoid pollution from linear motion, gravity estimation will be opportunistically used when the object stops or moves slowly. Thus, our task at hand is to solve this overdetermined system via sensor fusion.

3.2 Magnetometer + Gyroscope Fusion

Since gyroscopes measure changes in 3-DoF orientation, and magnetometers measure 2 DoFs of global orientation (which is $\boldsymbol{N}^G$) directly, the two sensors can be combined to better track (2 DoFs of) orientation. Thus, while gyroscope drift accumulates over time, the magnetometer can be used for recalibration (achieving better noise properties than either of the individual sensors). We use the complementary filter for this combining operation.

[2]In other words, roll and pitch angles are computed from gravity, while the yaw angle is computed from compass. This is a standard operation, which is also implemented by the Android SensorManager.getRotationMatrix() API: https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/hardware/SensorManager.java

To elaborate, assume that at current time t, the actual orientation of the object is $\boldsymbol{O}(t)$. $\boldsymbol{O}(t)$ is a 3X3 rotation matrix that rotates the object's GRF to LRF, and is not known to the object. Since the magnetometer measures the (constant) global 3D magnetic North vector $\boldsymbol{N}^G$ in its local reference frame (LRF), we can write its local measurement at time t as:

$$\boldsymbol{N}^L(t) = \boldsymbol{O}^{-1}(t)\boldsymbol{N}^G \tag{2}$$

Where $\boldsymbol{O}^{-1}(t)$ denotes the inverse of $\boldsymbol{O}(t)$. Of course, we don't know the actual orientation $\boldsymbol{O}(t)$, and our current *estimated* orientation, $\hat{\boldsymbol{O}}(t)$, derived from the gyroscope, may be erroneous. One way to check how large this error would be, is to use estimated orientation $\hat{\boldsymbol{O}}(t)$ to *infer* what the magnetometer should measure (as $\hat{\boldsymbol{N}}^L(t)$), i.e.,

$$\hat{\boldsymbol{N}}^L(t) = \hat{\boldsymbol{O}}^{-1}(t)\boldsymbol{N}^G \tag{3}$$

The difference between the inferred measurement $\hat{\boldsymbol{N}}^L(t)$, and the actual magnetometer measurement $\boldsymbol{N}^L(t)$, immediately reveals the drift in orientation[3], as illustrated in Figure 5.

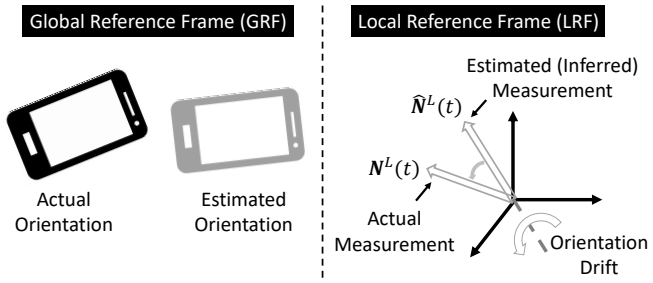

Figure 5: Magnetometer measures a globally constant vector, which helps correct the drift of orientation, reflected as the disparity between the inferred magnetometer measurement and the actual measurement.

■ **Now, how do we update orientation estimation using this disparity?** The key observation here is that the noise properties of magnetometers and gyroscopes are different, allowing for informed sensor fusion. Specifically, gyroscopes exhibit a long-term integration drift that grows with time[4], however, in the short term, it is quite accurate. Magnetometers, on the other hand, have short-term noise from environmental fluctuations and sensor imperfection, however, do not drift in the long run since they are always measuring the same global North vector and no integration is needed. To fuse the best of both sensors, we employ a *complementary filter* [6–8].

[3]To be precise, it reveals the drift in 2 of the 3 DoFs of orientation (the DoFs parallel to the 3D North vector direction).

[4]This is because the gyroscope is an inertial sensor that measures angular velocity in its local frame of reference, and has no opportunity to correct its own integration.

The complementary filter essentially computes a weighted combination of the two. Larger weight is assigned to the gyroscope, so that high frequency components are drawn from the gyroscope (since the gyroscope drifts less in the short term), and low frequency components from the magnetometer (since it is stable in the long run). The net output is a single orientation at each time instant – the estimate of 3D orientation in the GRF. Specifically, we first look for a delta rotation matrix ($\Delta\boldsymbol{R}$) that can align our *inferred* magnetometer measurement ($\hat{\boldsymbol{N}}^L(t)$) to the actual one ($\boldsymbol{N}^L(t)$):

$$\textit{Rotation Axis } \boldsymbol{e} = \hat{\boldsymbol{N}}^L(t) \times \boldsymbol{N}^L(t) \qquad \text{(Cross Product)}$$

$$\textit{Rotation Angle } \boldsymbol{\theta} = \measuredangle\left(\hat{\boldsymbol{N}}^L(t), \boldsymbol{N}^L(t)\right)$$

$$\Delta\boldsymbol{R} = \textit{AxisAngle2RotMat}\,(\boldsymbol{e}, \boldsymbol{\theta})$$

The inverse of this rotation matrix (i.e., $\Delta\boldsymbol{R}^{-1}$), when applied to our orientation estimation, will eliminate this disparity. Since we adopt a complementary filter design to reduce noise, we set a small coefficient α $(0 < \alpha \ll 1)$ for this operation, i.e.,

$$\Delta\boldsymbol{R}(\alpha) = \textit{AxisAngle2RotMat}(\boldsymbol{e}, \alpha\boldsymbol{\theta}) \tag{4}$$

And the updated orientation is

$$\hat{\boldsymbol{O}}_{\text{new}}(t) = \Delta\boldsymbol{R}^{-1}(\alpha) \cdot \hat{\boldsymbol{O}}_{\text{old}}(t) \tag{5}$$

As a technical detail, we are using axis-angle and rotation matrix representation of rotation, rather than Euler angles (roll/pitch/yaw), in order to avoid gimbal lock and ±180° ambiguity. In our implementation, we simply set α as 0.01. The final result from this filter is the convergence on 2 DoFs of orientation, while the 3rd DoF, i.e. rotation around the 3D magnetic North vector, is still tracked but prone to drift.

3.3 Implementation Details

We briefly mention a few details here.

Gyroscope Bias: IMU sensors are known to have bias (DC offset), among which gyroscope bias harms orientation estimation the most (because of integration). At the initial static moment where we perform one-time initialization of tracking, we also calibrate the bias by taking a time average of gyroscope readings, and remove the bias from subsequent gyroscope measurements.

Static Recalibration: Even though *MUSE* addresses the gyroscope drifting issue (due to integration) in 2 of the 3 DoFs of orientation, there is still 1 DoF whose error cannot be corrected during motion. Therefore, *MUSE* also opportunistically detects static or slow-moving opportunities (by looking at time windows in which the accelerometer measures roughly $9.8m/s^2$), if any, to address the drift in this dimension. Unlike A^3 which simply replaces current orientation

estimation with the one from gravity + North, we again use a complementary filter to update the estimation, which turns out to be more robust to accelerometer noise and have less false positives. We also leverage this opportunity to update our estimation of the 3D magnetic North anchor, $\boldsymbol{N}^G$.

Magnetometer Accuracy: MEMS magnetometers in mobile devices typically have lower resolution than accelerometers and gyroscopes. Luckily, we do not require as high resolution for magnetometers as for gyroscopes, because we are integrating the gyroscope readings but averaging the magnetometer readings (using the complementary filter). However, there might be ferromagnetic materials in the environment. Depending on the error distribution, in certain cases, it might be even better not to use the magnetometer. We evaluate the sensitivity of the algorithm to magnetic field fluctuations in the next section, and leave further investigation into the future work.

In sum, Algorithm 1 below shows the high-level pseudo code of *MUSE*'s orientation estimation algorithm.

Algorithm 1 *MUSE* Orientation Tracking

```
Opportunistically detect initial orientation and global 3D magnetic North anchor, using Equation (1)
while True do
    Integrate gyroscope to obtain new orientation
    if Accelerometer roughly measures 9.8m/s^2 then
        Recalibrate orientation estimation, and
        Update 3D magnetic vector estimation
    else
        Update orientation using Equation (3) - (5)
    end if
end while
```

We evaluate the accuracy of orientation estimation next, before proceeding to location tracking.

4 ORIENTATION EVALUATION

4.1 Experiment Design

■ **Platform and Test Scenarios:** *MUSE* uses the raw IMU data from a Samsung Galaxy S6 smartphone. It includes an InvenSense MPU6500 6-axis accelerometer + gyroscope, and a Yamaha YAS537 3-axis magnetometer. The same chips are also embedded in many other mobile and wearable devices, including other phone models (iPhone 6s, Amazon Fire Phone, Samsung Galaxy S5, Samsung Note 5), tablets (Kindle Fire HD), smartwatches (Samsung Gear Fit), VR headsets (HTC Vive, Oculus Rift), gaming controller (Oculus Touch, Steam Controller), etc. Tracking various motion patterns is of interest, for both *humans* and *things*. For humans, we begin with controlled activities, like pure linear motion, pure rotation, and their mixtures. Then, we generalize to real-world natural motions, including running, eating, basketball, gaming, etc. For these activities, we recruit volunteers to carry/wear the phone in different positions, such as in-hand, wrist, arm, and legs. We do not offer any guidance to volunteers; they perform the activities completely naturally. Finally, for object motion, we insert/paste the phone on various "things", including tennis racquets, soccer balls, bicycle wheels, etc.

■ **Metric:** Our main metric of interest is *3D orientation error* of the phone. Observe that this error need not be shown as separate errors around X, Y, and Z axes, respectively, but can be shown as a single orientation error (i.e., the minimal amount of rotation needed to align the estimated 3D orientation to the ground truth 3D orientation). Of course, this brings the question of determining the ground truth orientations.

■ **Ground Truth:** *MUSE* adopt A^3's technique of measuring orientation ground truth. We first start by placing the phone at a known orientation, using a printed protractor; then use the phone for the test motion or activity; and then bring back the phone to this known orientation. Since this end point is naturally a static moment for *MUSE*'s recalibration, we deliberately pause our algorithm and only use gyroscope integration for the last few seconds of motion. This ensures that the true motion tracking error is measured (without an artificial orientation reset at the end). Finally, since many of our motion tracking sessions will be long (5+ minutes), we will periodically bring the phone to the ground-truth orientation, pause, and then continue natural motion again. However, we will not use these artificial pauses for recalibrating 3D orientation, but only to measure the ground truth at intermediate points during motion. This will offer insights into the intermediate moments while motion is in progress.

■ **Comparison Baselines:** Last but not least, we will compare *MUSE*'s performance against 3 other techniques. **(1) A3** from MobiCom 2014; **(2) GyroOnly**, indicating rotation estimation from 3-DoF gyroscope integration alone, with no gravity or magnetometer; and **(3) ComplemFilter**, indicating the traditional use of complementary filter for IMU sensor fusion, which constantly combines gyroscope integration with the estimation from gravity + North. Across all these cases, the algorithms are executed in MATLAB, using the same IMU data supplied by the *SensorManager* API from Android.

4.2 Results

We begin with the discussion on basic (controlled) motions, and then evaluate natural activities and gestures.

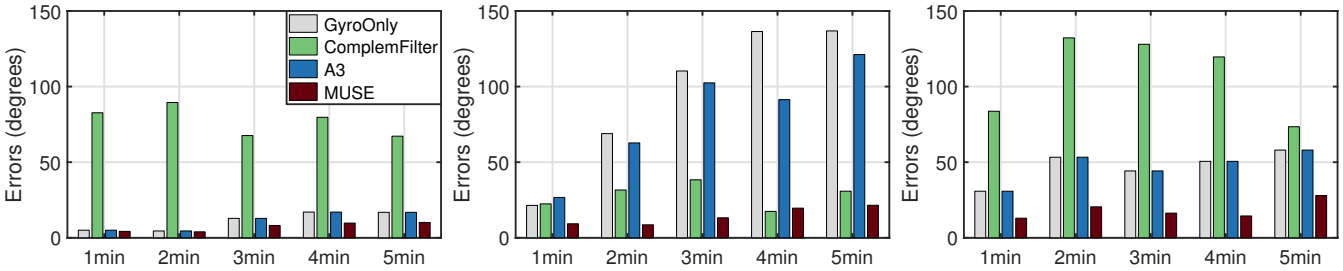

Figure 6: Orientation tracking error for three basic controlled motions: (a) pure translation, (b) pure rotation, and (c) mixtures of translational and rotational motions.

Basic Controlled Motions

Figure 6 plots 3D orientation error for pure translation, pure rotation, and mixtures of translational and rotational motions. For pure translation, the phone is continuously moved in different straight-line directions (not necessarily horizontal or vertical, but other possible diagonal lines). For pure rotation, the phone is located at roughly the same position, but rotated around various axes (not just X, Y, and Z). For instance, the phone could be rotated around an axis defined by the vector $\vec{V} = \vec{X} + \vec{Y}$). Finally, for motion mixtures, we perform random actions involving both linear and rotational motion.

MUSE consistently performs well, while other techniques falter in some scenario or the other. Complementary filter, for instance, gets affected in the presence of translational motion since gravity is polluted. Gyroscope integration incurs error when phone rotation is dominant. Finally, A^3 is relatively better but still considerably worse than *MUSE* due to the lack of static moments for resetting orientation. In other words, the strong trust on magnetometer serves *MUSE* well in estimating the phone orientation.

Figure 7 compares the overall CDF of 3D orientation error across all controlled scenarios. While A^3 performs better than GyroOnly and ComplemFilter, *MUSE* exhibits a consistent improvement over A^3 (3.5X gain at median and 4X gain at 90th percentile), closing in the gaps for IMU based motion tracking.

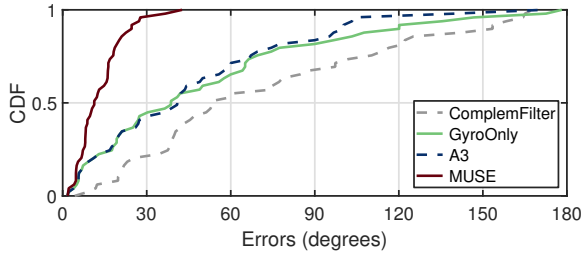

Figure 7: Overall orientation error CDF for controlled motions.

Natural Human and Object Motions

Figure 8 plots orientation error across various natural movements of humans and objects. For instance, an object motion like "Tennis" refers to the phone taped on the strings of the tennis racquet and a user pretending to play with it; the "Ball" refers to a user playing with a ball with a phone tightly pasted to it. Evident from Figure 8, *MUSE* outperforms the other methods almost across all activities. A^3 is comparable when the motion is naturally slow such that the accelerometer pollution is not excessive (e.g., eating); or when the motion has natural pauses (e.g., 3D mouse); but falls short when the motion is continuous and without stops. On average, *MUSE* achieves 2.9X smaller orientation error than A^3. As for GyroOnly and ComplemFilter, the mixture of linear and rotational motion in natural activities affects their performance.

Finally, Figure 9 plots a sample trace of the 3D orientation error over time to demonstrate how the error can grow if A^3 is unable to find adequately frequent pause moments, or makes mistakes in identifying them. In contrast, *MUSE* maintains a low error, mainly due to magnetometer noise and the gyroscope's drift.

4.3 When Will MUSE Fail?

■ Rotation Only in 3rd DoF

Utilizing the 3D magnetic North vector as an anchor, *MUSE* provides an overdetermined system in 2 of the 3 DoFs of orientation. This means that *MUSE* will not be useful, if the object's rotational motion happens to be only in the 3rd DoF, i.e. the object is exactly rotating around the global anchor direction (and will not change its axis of rotation thereafter). Of course, this rarely happens in practice, and even if it does, any static recalibration opportunity will mitigate this problem.

■ No Opportunity for Initialization

While methods such as A^3 rely on frequent "pauses", *MUSE* gets rid of this assumption but still needs one static moment

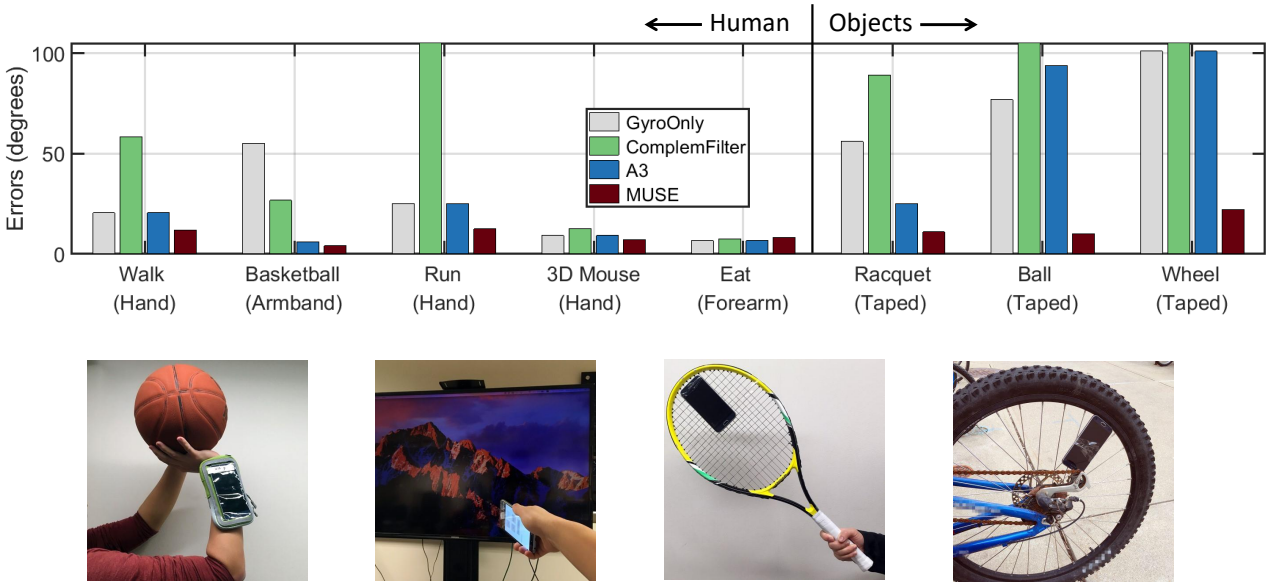

Figure 8: (a) Orientation error across natural activities, including both human (left) and object (right) motions. (b) The phone position for some of the motion experiments.

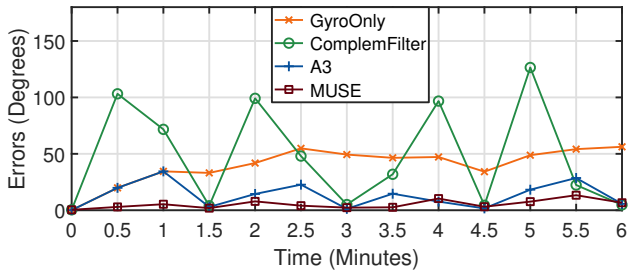

Figure 9: A sample trace of orientation error over time for different techniques. Intermediate ground-truth probing happens every half a minute, and the device's motion pauses on purpose every 1.5 minutes (for re-calibration).

to bootstrap the tracking. If there is not even a single pause or slow-moving moment from the beginning, then *MUSE* will not be able to find an opportunity to compute initial orientation and the 3D magnetic North anchor.

■ Ferromagnetic Materials

Since *MUSE* relies on the magnetic North vector as a global anchor, it would degrade in performance when the magnetic interference is strong (which is also the case for other systems). While deeper treatment is necessary in the future, we bring *MUSE* to more challenging environments to test its sensitivity to ferromagnetic materials.

Figure 10 shows the orientation tracking accuracy when we run *MUSE* at different places, ranging from outdoor and large indoor open space (with least interference), to crowded engineering buildings and labs with lots of computers and cables (with strongest interference). Each dot in the figure represents one trial, and its X value describes how fluctuated the magnetic field is. We measure the X value by moving and rotating the device around in this area, and taking the standard deviation of the magnitudes of magnetometer measurements. Clearly, *MUSE*'s performance decreases as the magnetic field density variation increases. Techniques such as magnetic field profiling may help mitigate this issue – a topic we leave to future work.

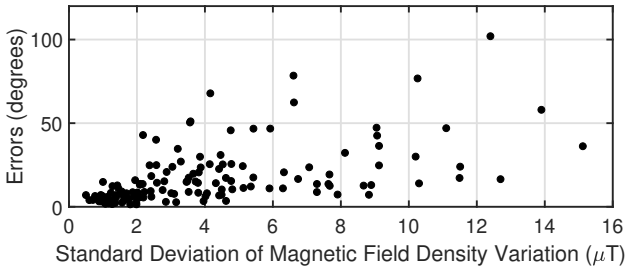

Figure 10: *MUSE*'s tracking accuracy as it runs at different places with varying levels of magnetic field fluctuations.

With the *MUSE* orientation tracking in place, we now turn to tracking the location of an IMU sensor.

5 PART II: LOCATION ESTIMATION

The general problem of 3D location tracking is much harder compared to 3D orientation. This is fundamentally because there is *no notion of a "location anchor", precluding the IMU device from knowing how far it has diverged from truth.* Thus, any location error continues to accumulate without bounds (unlike orientation where gravity and magnetic North helps to reset the error). To elaborate, consider the steps for computing location at any given time: (1) the accelerometer data must be projected to the global reference frame (GRF) using the current estimate of orientation; (2) gravity must be removed; and (3) only the linear motion component must be double integrated. If the GRF projection or gravity removal is slightly imperfect, or if the accelerometer has hardware noise, the double integration operation will amplify the error over time. Again, there is no way to reset this error.

While this is true for general 6-DoF motion (i.e., the object can move and rotate freely in any manner), opportunities emerge when the motion has restrictions. A simple example of restriction is as follows: a human's elbow location must always lie on the surface of a sphere centered at the shoulder. If such restrictions can be modeled, then it could serve as a framework to evaluate our location estimates. For instance, if the elbow's estimated location diverges from the spherical surface around the shoulder, then we have the opportunity to recalibrate. This is the core insight for the rest of this paper. We will show that motion models can help to overdetermine the location estimates (much like orientation), and thereby, *location and orientation can be jointly estimated* for such movements. Thus, *MUSE* will not compute orientation first followed by location, rather, feed all the available IMU information and motion models into a Bayesian Particle Filter framework, to simultaneously extract out 3D location and 3D orientation.

To ground our technique in a concrete application, we will track a user's wrist location using IMU data from her smartwatch, and motion models of the human arm. *ArmTrak* [12] in MobiSys'16 adopts arm-motion models to solve this problem, however, uses A^3 to estimate orientation first, and then infers location. Further, *ArmTrak* relies on full offline data to perform motion tracking. *MUSE* is different in its core magnetometer based orientation module, as well as in the joint estimation of orientation and location. Finally, we believe our technique generalizes to many other motion models.

5.1 Arm Motion Model

We adopt a classical limb motion model [12–14], illustrated in Figure 11. Assuming the torso is static, the watch (on wrist) motion is actually controlled by shoulder and elbow joints that lie in a 5-DoF space. The 5-DoF motion can either

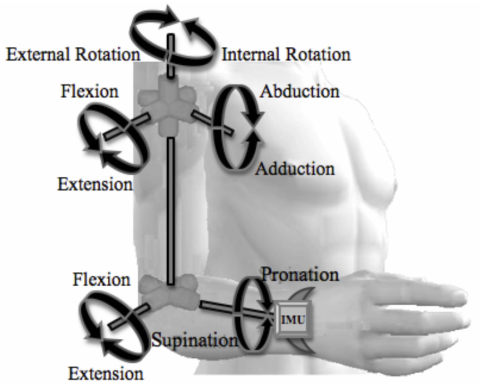

Figure 11: Arm motion model borrowed from [13].

be viewed as 3 DoFs from upper arm rotation + 2 DoFs from forearm rotation (Figure 11), or as 3 DoFs from watch orientation + 2 DoFs from watch location (due to fixed forearm and upper arm lengths). Moreover, due to the nature of joints, muscles, and ligaments in humans, arm motion is restricted to certain ranges, called *range of motion* (RoM) [12, 14]. For instance, a human cannot twist her forearm by 360°, and elbows cannot bend beyond 180°. We use the same RoM as being used in *ArmTrak*. DoF + RoM together determine the motion model.

5.2 System Overview

Given the coupling between orientation and location, jointly estimating them is the correct approach to fully utilize IMU data and the motion model. However, the key issue is the following: On one hand, jointly tracking the space of 5 unknowns (state variables) is prone to divergence, while also being computation heavy. On the other hand, separately estimating orientation and location, such as *ArmTrak*, reduces complexity but at the expense of accuracy.

To mitigate this issue, we propose a different approach. Figure 12 shows our system design. We first use *MUSE* to estimate the orientation of the IMU sensor. Among the 3 DoFs of orientation, 2 of them will be drift-free. The remaining 1-DoF orientation, together with the 2 DoFs of location, is to be estimated from the 3 DoFs of the accelerometer. We then ask the following question: *what sequence of watch orientations and locations will cause gravity and linear motion to (vectorially) add up in a way that matches the accelerometer measurements?* This motivates our particle filter design, explained next.

5.3 Joint Tracking via Particle Filter

Figure 13 shows a simple, one-step example to illustrate the basic idea. The smartwatch was previously at location $\boldsymbol{L}(t-1)$, and now arrives at location $\boldsymbol{L}(t)$ at time t. We have some

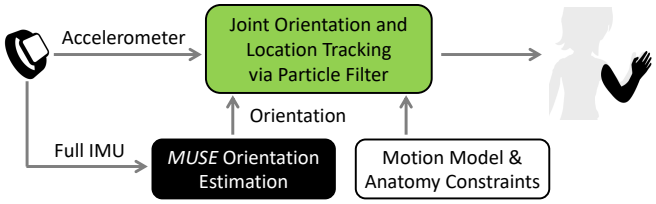

Figure 12: System overview for joint location and orientation tracking. Orientation estimation from *MUSE*, as well as the accelerometer data, is sent to the particle filter for joint tracking.

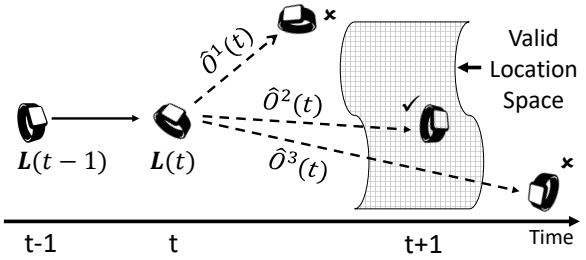

Figure 13: A illustrative example of how incorrect orientations lead to bad location estimations.

uncertainty over our current orientation estimation, $\hat{O}(t)$, most likely due to gyroscope integration drift that causes 1-DoF errors (i.e. uncertainty on the rotation that is around the magnetic North direction). If we aim to track location, then different orientation estimates, say $\hat{O}^1(t)$, $\hat{O}^2(t)$ and $\hat{O}^3(t)$, will project accelerometer data differently. Assume $\hat{O}^2(t)$ is the correct orientation – it will project accelerometer data correctly, using which the location estimation will also be correct, and the location will always stay in the *valid location space* (i.e., satisfying 5-DoF space model and arm range-of-motion constraints). On the other hand, a wrong orientation, for example $\hat{O}^1(t)$, will make mistakes in projecting the accelerometer, eventually making the estimated location fall outside of the valid location space. In sum, due to lower dimensionality of the valid location space, we now have the opportunity to understand and possibly correct orientation drift from accelerometer, even under motion.

Of course, the above is an over-simplified example. In reality, the past locations are not known a priori and have to be estimated, and the valid location space depends on smartwatch orientation. Therefore, given this highly non-linear state space, we model each possible ⟨*orientation*, *location*⟩ trajectory as a particle in the particle filter. Our hope is that, particles with wrong orientations will exhibit diminishing weights (or probabilities) and eventually get removed. Particles that are left behind are more likely to be good joint estimators of both orientation and location. We detail the particle filter construction below.

■ **State Space:** We define each particle as a vector, composed of the 3 recent watch location estimates ($\boldsymbol{L}$), as well as an angle that models the orientation drift around magnetic North vector at the previous time (δ).

$$(i\text{-th particle})\ \boldsymbol{X}^i = \left[\boldsymbol{L}^i(t)\ \boldsymbol{L}^i(t-1)\ \boldsymbol{L}^i(t-2)\ \delta^i(t-1)\right]$$

■ **Weight Updating:** A good estimate of the state should be such that (1) it lies in the *valid location space*, and (2) it matches the accelerometer measurements well. Therefore, we can evaluate how good each estimate (particle) is, by comparing the accelerometer sensor readings with the "inferred" acceleration from the particle's location trajectory. In physics, (for the i-th particle) the global acceleration of the watch can be approximated by:

$$\frac{\mathrm{d}^2\boldsymbol{L}^i(t-1)}{\mathrm{d}t^2} \approx \frac{\boldsymbol{L}^i(t) - 2\boldsymbol{L}^i(t-1) + \boldsymbol{L}^i(t-2)}{\Delta t^2} \tag{6}$$

On the other hand, we get global acceleration measurement, by projecting accelerometer data with estimated orientation and removing gravity afterwards. Specifically, at each time *MUSE* outputs an estimated orientation, $\hat{\boldsymbol{O}}(t-1)$, which may have a drift around the magnetic North vector. Each particle tries to calibrate that drift by adding its own drift estimation $\delta^i(t-1)$:

$$\hat{\boldsymbol{O}}^i(t-1) = \Delta R(\delta^i(t-1)) \times \hat{\boldsymbol{O}}(t-1)$$

After that, each particle uses its own calibrated orientation, $\hat{\boldsymbol{O}}^i(t-1)$, to estimate the global linear acceleration (by projection and removing gravity):

$$\frac{\mathrm{d}^2\boldsymbol{L}^i(t-1)}{\mathrm{d}t^2} = \hat{\boldsymbol{O}}^i(t-1) \cdot accel(t-1) - \left[0 \quad 0 \quad \text{gravity}\right]^T \tag{7}$$

Observe Equation (6) and Equation (7): one is (inferred) global acceleration from the particle's location triple, and the other one is global acceleration from the particle's (drift-corrected) orientation projection. A converging system state should have minimal difference between these two terms. We model this by assigning weights to particles, where the weight is a zero-mean Gaussian probability density function on their difference.

■ **Re-sampling:** We resample at each step so that particles are more concentrated to the ones with higher probability (i.e., better fitted to accelerometer measurements). Particles with higher weights are likely to get multiple copies, allowing the system to focus on the more-likely state trajectories.

■ **Prediction:** As $t \rightarrow t+1$, the particle filter updates itself with the following two steps: *First*, it takes the averaged drift correction across all particles, $\overline{\delta}(t-1)$, and use it to calibrate *MUSE* output, before it proceeds to the next timestamp. All

particles will thereafter reset their drift correction, $\delta^i(t-1)$, to a new small Gaussian random number. *Second*, as time proceeds, the location triple should simply be updated as well.

$$\begin{aligned} \boldsymbol{L}^i(t-2) &\leftarrow \boldsymbol{L}^i(t-1) \\ \boldsymbol{L}^i(t-1) &\leftarrow \boldsymbol{L}^i(t) \\ \boldsymbol{L}^i(t) &\leftarrow \text{(A random point from valid location space)} \end{aligned}$$

Note that the new watch location, $\boldsymbol{L}^i(t)$, is a point randomly selected from the valid location space in order to preserve diversity of the system. For each particle, the valid location space is conditioned on its predicted orientation $\hat{\boldsymbol{O}}^i(t)$, which we have predicted using *MUSE*'s output $\hat{\boldsymbol{O}}(t)$ and calibrated at the prediction step. We also restrict $\boldsymbol{L}^i(t)$ not to be far away from $\boldsymbol{L}^i(t-1)$.

To improve the speed of the particle filter, we aggregate every K samples together, so the particles are updated and resampled every $K\Delta t$ seconds. K serves as a tuning parameter to balance computation and latency, and we empirically set $K\Delta t$ to 0.1s, corresponding to 10Hz system update frequency. We are now ready to evaluate the accuracy of this tracking algorithm.

6 LOCATION EVALUATION

6.1 Experiment Design

■ **Platform and Test Scenarios:** We run our algorithm using raw IMU data collected from Samsung Gear Live smartwatches, which include an InvenSense MP92M 9-axis IMU chip. The data is processed on MATLAB in an online, streaming manner. We hire 5 student volunteers and ask them to wear the watch on wrist. To ensure the accuracy of the arm motion model, we measure the upper arm and forearm lengths of each user.

Since arm model and range-of-motion are defined with respect to the torso, we also insert a smartphone into each user's pant pocket, for roughly estimating their torso facing direction, and therefore aligning their torso reference frame (TRF, as shown in Figure 14) with global reference frame (GRF).

Our arm motion model is most accurate when the user's body is not moving, otherwise torso motion will pollute smartwatch IMU data. We divide the full experiment cycle into 3 sessions: arm motion under (1) static body, (2) rotating body (i.e., only the facing direction is changing), and (3) both walking and rotating body. The volunteers were asked to perform both random and natural gestures, such as eating, smoking, lecturing, walking and pointing, ball room dancing, etc.

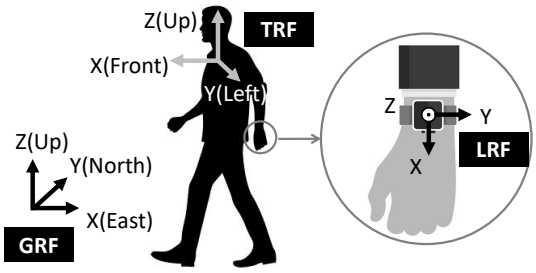

Figure 14: Various reference frames: local, torso, and global. Smartwatch IMU measures motion in local reference frame (LRF); arm model and range-of-motion are defined with respect to the torso reference frame (TRF).

■ **Metric and Ground Truth:** Since it is difficult to obtain the ground truth of smartwatch orientation, we evaluate the accuracy of location tracking. Location error is computed via two separate infrastructures. The first is a VICON-based motion capture system with 8 infrared (IR) cameras on the ceiling and small IR markers placed on the humans. The IR stickers help reflect IR light, allowing the cameras to accurately localize 3D location. The second method is through a Kinect 2.0, placed closer to the human (clearly no markers were needed in this case). In both cases, we prevent movements in the background to avoid interference with the ground-truth computation. Unfortunately, we were unable to use VICON at the end since the IR markers often got occluded due to arm and body motion, resulting in windows of missing data. Hence, all our round truths are from Kinect.

6.2 Results

■ **Overall Accuracy**

Figure 15 shows the overall CDF of watch location error in the global reference frame, under 3 different background body motions. Highest accuracy is achieved when the torso is static, with median error of 8.8cm. As the body starts rotating, the error increases slightly to 9.7cm, mainly due to inaccurate facing direction estimation from the phone. As the body also starts moving around, the motion model becomes less applicable, and the accuracy further degrades to 13.0cm. As a comparison, *ArmTrak* achieves an accuracy of 9.2cm for static torsos, using offline decoding with future data.

■ **Accuracy across Users**

Figure 16 demonstrates the consistency of tracking across 5 different users. None of the users were trained in any way, and were asked to move their hand in a completely unsupervised manner. Their arm lengths are also shown in Table 1. Evidently, there is no strong relationship between the arm length and the watch's location error. However, as expected,

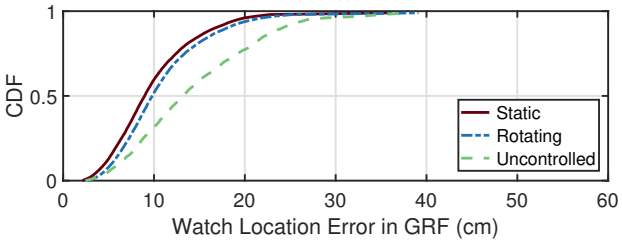

Figure 15: Overall CDF of watch location error in the global reference frame (which is the aligned torso reference frame).

the performance variance across users grows with richer body motion. The reason is rooted in each user having her unique body motion pattern, bringing different errors to the system.

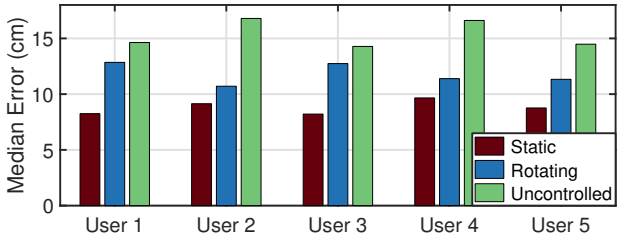

Figure 16: Accuracy across 5 users, under various body motion.

User ID	1	2	3	4	5
Arm Length (cm)	53.2	50.4	48.7	51.5	51.9

Table 1: Arm lengths of 5 users.

■ Does the Error Diverge over Time?
Figure 17 shows how watch location error fluctuates over time for a randomly picked trajectory. The error is large at first, as the system is unaware of the watch's starting location. Once the arm starts moving, the overall error goes down and does not drift over time. This is a desirable property for inertial tracking systems, which are otherwise known to diverge due to error accumulation.

■ Tracking across Applications
As a final experiment, we also asked volunteers to perform certain arm gestures as if they were in different applications scenarios. These applications include: (1) Eating: a user performs eating gestures while standing still. (2) Gym: a user performs common gym gestures such as dumbbell lifting, pull-backs, and boxing. (3) MR: medical rehabilitation, where a user performs reach-and-grasp tasks, moves objects, etc., similar to how stroke patients are treated in the months after the surgery. (4) AR: augmented reality, where a user casually walks and turns around in a (museum) room and points at

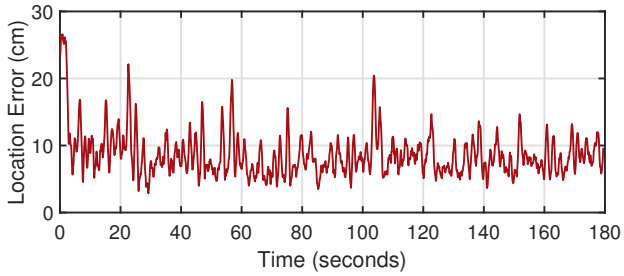

Figure 17: An example of tracking accuracy over time: the error does not diverge.

certain objects. We envision information popping up on the watch based on the object she points at. (5) Basketball: where a user bounces the ball while moving briskly among other users.

Figure 18 plots the results for each application, in increasing order of complexity. The bars are those of 2 users that performed worst from the set of 5 – again a conservative result. Obviously, the performance is better at tracking simple gestures (such as eating, gym) with relatively modest level of background body motion. Results degrade with increasingly complicated motions, and exhibit a sudden rise with highly aggressive movements in basketball.

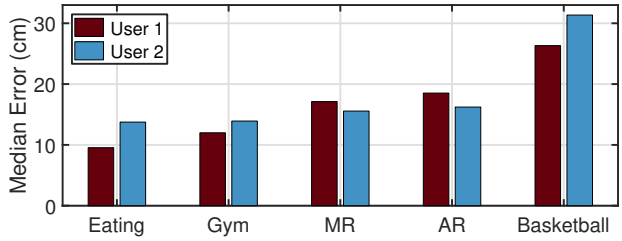

Figure 18: Watch location error in applications.

7 POINTS OF DISCUSSION

MUSE leaves room for further investigation, as discussed briefly in the interest of space:

(1) Optimality: We have not been able to comment on the optimality of IMU based tracking. This needs a deeper signal processing treatment via models of random sensor noise (and bias). Of course, when ignoring noise, we know that 3D orientation is solvable given at least 5 DoFs of information (2 from magnetometer and 3 from gyroscope). Location is also solvable, although more sensitive since the system is just adequately determined (3 DoFs of accelerometer for 3D location). For real systems, however, the interplay of hardware noise and restrictions of motion models will together determine the system's error. We leave this analysis to future work.

(2) Running in Real Time: Part I runs a light-weight complementary filter, which has less computational complexity than related works that can already run on mobile and wearable devices in real time [11, 15, 16]. We ran Part II, the particle filter, on a quad-core student desktop. The algorithm outputs results at the rate of 10Hz (depending on the particle filter update rate). The desktop needs 0.06 seconds to process every 0.1 second of raw data. While the computation cannot be afforded by a smartphone locally, it should be easy to process it in the cloud, or an edge-computer. Given our bandwidth requirement is a minimal 20kb/s, we expect *MUSE* to operate in near-real time, with sub-second end-to-end latencies.

(3) Magnetic Interference: In Part II, we performed our experiments in an indoor office fitted with Kinect and VICON ground truth systems. While the magnetic fluctuations in this room were modest, it is possible that other locations (e.g., factories) could experience severe fluctuations. Under heavy fluctuations, it may be possible to perform joint arm tracking and magnetic field profiling. This leaves another dimension of improvement in the area of IMU tracking.

8 RELATED WORK

■ **IMU Orientation Estimation:** As mentioned earlier, this problem has been well studied in aerodynamics and robotics, and various algorithms have been proposed to derive efficient sensor fusion algorithms under specific error models [9, 10]. While some of these algorithms [4, 5, 11, 17–21] also use all the available information of the magnetometer, they assume that the object's motion is slow or has intermittent stops. Table 2 summarizes some of the important related works, and classifies them based on their sensor fusion techniques, and the key assumptions they have made. These assumptions include:

(A): The linear motion is slow so that the average of accelerometer is gravity.
(B): The rotational motion is slow and the error model is Gaussian, in order to preserve the linearity of the system model.
(C): The motion has frequent pauses (static moments) for resetting the gravity estimation.

However, these assumptions may easily break down for continuous motion from human-wearable devices.

■ **IMU Location Tracking:** Prior research has successfully used IMU sensors for activity recognition and classification [22–26], such as eating, smoking, typing, etc. However, tracking IMU's exact location, also known as "IMU dead reckoning", is a much more challenging task. Past works [27–29] have been able to track coarse-grained IMU location for short-time human walking motion, by counting steps and estimating step lengths using the IMU sensor. [14, 30–32] have proposed to track locations of IMUs attached to human limbs. However, they attach multiple IMUs, one for each rigid-body component, essentially tracking the orientation (rather than location) of the IMUs. Our work tracks the location of a single wrist-worn IMU using arm motion models. Compared with [12] which relies on full future data to perform offline Viterbi decoding for IMU location, we perform joint location-orientation tracking, and achieve better performance even with online decoding.

Related Works	Sensor Fusion Techniques	Key Assumptions
[5–8, 33, 34]	Complementary Filter	(A)
[17, 35–38]	Kalman Filter	(A) (B)
[39, 40]	Kalman Filter	(B) (C)
[3, 4, 18, 19, 41–45]	EKF Filter	(A) (B)
[2]	EKF Filter	(B) (C)
[20, 21, 46]	UKF Filter	(A)
[11]	Opportunistic Replacement	(C)
[47] (Android APIs)	Gravity+Compass Only	(A)

Table 2: Related works on IMU orientation tracking, classified based on their sensor fusion techniques and key assumptions made.

■ **Other Tracking Modalities:** Many other modalities can track the position and motion of objects, including IR technology [48, 49], computer vision [50, 51], wireless sensing [52–55], RFID [56, 57], visible light [58], acoustics [59–61], etc. However, the core inertial nature of the MEMS IMU sensors presents unique challenges distinct from other sensing modalities.

9 CONCLUSION

This paper shows improvements to orientation tracking by recognizing that magnetometers, unlike accelerometers, are unpolluted by object motion. Then, the paper shows that motion models can be employed to restrict the divergence of location estimation, ultimately allowing for a joint estimate of both 3D location and 3D orientation.

ACKNOWLEDGMENTS

We sincerely thank the anonymous shepherd and reviewers for their insightful comments and suggestions. We are also grateful to Intel, Google, Qualcomm, HP and NSF (grant NSF 1719337) for partially funding this research.

REFERENCES

[1] "Grush smart oral care," https://www.grushgamer.com/.

[2] Billur Barshan and Hugh F Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[3] João Luís Marins, Xiaoping Yun, Eric R Bachmann, Robert B McGhee, and Michael J Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. IEEE, 2001, vol. 4, pp. 2003–2011.

[4] Angelo M Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[5] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–7.

[6] Hyung-Jik Lee and Seul Jung, "Gyro sensor drift compensation by Kalman filter to control a mobile inverted pendulum robot system," in *Industrial Technology, 2009. ICIT 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[7] Mark Euston, Paul Coote, Robert Mahony, Jonghyuk Kim, and Tarek Hamel, "A complementary filter for attitude estimation of a fixed-wing uav," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 340–345.

[8] Eric R Bachmann, Robert B McGhee, Xiaoping Yun, and Michael J Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," in *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 2001, pp. 9–16.

[9] John L Crassidis, F Landis Markley, and Yang Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[10] Haiyang Chao, Calvin Coopmans, Long Di, and YangQuan Chen, "A comparative evaluation of low-cost IMUs for unmanned autonomous systems," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*. IEEE, 2010, pp. 211–216.

[11] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 605–616.

[12] Sheng Shen, He Wang, and Romit Roy Choudhury, "I am a smartwatch and i can track my user's arm," in *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, 2016.

[13] Mahmoud El-Gohary, Sean Pearson, and James McNames, "Joint angle tracking with inertial sensors," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 1068–1071.

[14] M. El-Gohary and J. McNames, "Shoulder and elbow joint angle tracking with inertial sensors," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 9, pp. 2635–2641, 2012.

[15] Jarosław Gośliński, Michał Nowicki, and Piotr Skrzypczyński, "Performance comparison of EKF-based algorithms for orientation estimation on android platform," *IEEE Sensors Journal*, vol. 15, no. 7, pp. 3781–3792, 2015.

[16] Michał Nowicki, Jan Wietrzykowski, and Piotr Skrzypczyński, "Simplicity or flexibility? complementary filter vs. EKF for orientation estimation on mobile devices," in *Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on*. IEEE, 2015, pp. 166–171.

[17] Nguyen Ho Quoc Phuong, Hee-Jun Kang, Young-Soo Suh, and Young-Sik Ro, "A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 859–876, 2009.

[18] Ezzaldeen Edwan, Jieying Zhang, Junchuan Zhou, and Otmar Loffeld, "Reduced DCM based attitude estimation using low-cost IMU and magnetometer triad," in *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on*. IEEE, 2011, pp. 1–6.

[19] Håvard Fjær Grip, Thor I Fossen, Tor A Johansen, and Ali Saberi, "Globally exponentially stable attitude and gyro bias estimation with application to gnss/ins integration," *Automatica*, vol. 51, pp. 158–166, 2015.

[20] M Romanovas, L Klingbeil, M Trachtler, and Y Manoli, "Efficient orientation estimation algorithm for low cost inertial and magnetic sensor systems," in *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE, 2009, pp. 586–589.

[21] Benoit Huyghe, Jan Doutreloigne, and Jan Vanfleteren, "3d orientation tracking based on unscented Kalman filtering of accelerometer and magnetometer data," in *Sensors Applications Symposium, 2009. SAS 2009. IEEE*. IEEE, 2009, pp. 148–152.

[22] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[23] Y. Dong, A. Hoover, J. Scisco, and E. Muth, "A new method for measuring meal intake in humans via automated wrist motion tracking," *Applied psychophysiology and biofeedback*, vol. 37, no. 3, pp. 205–215, 2012.

[24] A. Parate, M. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "RisQ: Recognizing smoking gestures with inertial sensors on a wristband," in *ACM MobiSys*, 2015.

[25] H. Wang, T. T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *ACM MobiCom*. ACM, 2015, pp. 155–166.

[26] Chao Xu, Parth H Pathak, and Prasant Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 2015, pp. 9–14.

[27] Wonho Kang and Youngnam Han, "Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors journal*, vol. 15, no. 5, pp. 2906–2916, 2015.

[28] Ngoc-Huynh Ho, Phuc Huu Truong, and Gu-Min Jeong, "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone," *Sensors*, vol. 16, no. 9, pp. 1423, 2016.

[29] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.

[30] H. Zhou, H. Hu, and Y. Tao, "Inertial measurements of upper limb motion," *Medical and Biological Engineering and Computing*, vol. 44, no. 6, pp. 479–487, 2006.

[31] H. Zhou and H. Hu, "Upper limb motion estimation from inertial measurements," *International Journal of Information Technology*, vol. 13, no. 1, pp. 1–14, 2007.

[32] A. G. Cutti, A. Giovanardi, L. Rocchi, A. Davalli, and R. Sacchetti, "Ambulatory measurement of shoulder and elbow kinematics through inertial and magnetic sensors," *Medical & biological engineering & computing*, vol. 46, no. 2, pp. 169–178, 2008.

[33] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[34] Sebastian Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, pp. 113–118, 2010.

[35] Seanglidet Yean, Bu Sung Lee, Chai Kiat Yeo, and Chan Hua Vun, "Algorithm for 3d orientation estimation based on Kalman filter and gradient descent," in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*. IEEE, 2016, pp. 1–6.

[36] Vadim Bistrov, "Performance analysis of alignment process of MEMS IMU," *International Journal of Navigation and Observation*, vol. 2012, 2012.

[37] Lu Lou, Xin Xu, Juan Cao, Zhili Chen, and Yi Xu, "Sensor fusion-based attitude estimation using low-cost MEMS-IMU for mobile robot navigation," in *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*. IEEE, 2011, vol. 2, pp. 465–468.

[38] Wei Li and Jinling Wang, "Effective adaptive Kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems," *The Journal of Navigation*, vol. 66, no. 1, pp. 99–113, 2013.

[39] Z Ercan, V Sezer, H Heceoglu, C Dikilitas, M Gokasan, A Mugan, and S Bogosyan, "Multi-sensor data fusion of DCM based orientation estimation for land vehicles," in *Mechatronics (ICM), 2011 IEEE International Conference on*. IEEE, 2011, pp. 672–677.

[40] David Jurman, Marko Jankovec, Roman Kamnik, and Marko Topič, "Calibration and data fusion solution for the miniature attitude and heading reference system," *Sensors and Actuators A: Physical*, vol. 138, no. 2, pp. 411–420, 2007.

[41] Rong Zhu, Dong Sun, Zhaoying Zhou, and Dingqu Wang, "A linear fusion algorithm for attitude determination using low cost MEMS-based sensors," *Measurement*, vol. 40, no. 3, pp. 322–328, 2007.

[42] Eric Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter," in *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*. IEEE, 1996, pp. 185–194.

[43] Rodrigo Munguia and Antoni Grau, "Attitude and heading system based on EKF total state configuration," in *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 2147–2152.

[44] Demoz Gebre-Egziabher, Roger C Hayward, and J David Powell, "Design of multi-sensor attitude determination systems," *IEEE Transactions on aerospace and electronic systems*, vol. 40, no. 2, pp. 627–649, 2004.

[45] N Shantha Kumar and T Jann, "Estimation of attitudes from a low-cost miniaturized inertial platform using Kalman filter-based sensor fusion algorithm," *Sadhana*, vol. 29, no. 2, pp. 217–235, 2004.

[46] Hector Garcia De Marina, Fernando J Pereda, Jose M Giron-Sierra, and Felipe Espinosa, "Uav attitude estimation using unscented Kalman filter and triad," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4465–4474, 2012.

[47] "Position sensors, android developers," https://developer.android.com/guide/topics/sensors/sensors_position#sensors-pos-orient.

[48] "Motion capture systems - vicon," http://vicon.com/.

[49] "Motion capture systems - optitrack," http://optitrack.com/.

[50] "Microsoft kinect," https://dev.windows.com/en-us/kinect.

[51] "Intel realsense technology," http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html.

[52] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller, "3d tracking via body radio reflections," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 317–329.

[53] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 27–38.

[54] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand, "Capturing the human figure through a wall," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 219, 2015.

[55] Deepak Vasisht, Swarun Kumar, and Dina Katabi, "Decimeter-level localization with a single wifi access point.," in *NSDI*, 2016, vol. 16, pp. 165–178.

[56] Longfei Shangguan, Zimu Zhou, and Kyle Jamieson, "Enabling gesture-based interactions with objects," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 239–251.

[57] Teng Wei and Xinyu Zhang, "Gyro in the air: tracking 3d orientation of batteryless internet-of-things," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 55–68.

[58] Tianxing Li, Qiang Liu, and Xia Zhou, "Practical human sensing in the light," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 71–84.

[59] Wenguang Mao, Jian He, and Lili Qiu, "Cat: high-precision acoustic motion tracking," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 69–81.

[60] Sangki Yun, Yi-Chao Chen, and Lili Qiu, "Turning a mobile device into a mouse in the air," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 15–29.

[61] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1515–1525.