I am a Smartphone and I can Tell my User's Walking Direction

Nirupam Roy
University of Illinois (UIUC)
nroy8@illinois.edu

He Wang
University of Illinois (UIUC)
hewang5@illinois.edu

Romit Roy Choudhury
University of Illinois (UIUC)
croy@illinois.edu

ABSTRACT

This paper describes *WalkCompass*, a system that exploits smartphone sensors to estimate the direction in which a user is walking. We find that several smartphone localization systems in the recent past, including our own, make a simplifying assumption that the user's walking direction is known. In trying to relax this assumption, we were not able to find a generic solution from past work. While intuition suggests that the walking direction should be detectable through the accelerometer, in reality this direction gets blended into various other motion patterns during the act of walking, including up and down bounce, side-to-side sway, swing of arms or legs, etc. Moreover, the walking direction is in the phone's local coordinate system (e.g., along Y axis), and translation to global directions, such as 45° North, can be challenging when the compass is itself erroneous. WalkCompass copes with these challenges and develops a stable technique to estimate the user's walking direction within a few steps. Results drawn from 15 different environments demonstrate median error of less than 8 degrees, across 6 different users, 3 surfaces, and 3 holding positions. While there is room for improvement, we believe our current system can be immediately useful to various applications centered around localization and human activity recognition.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques; C.5.3 [**Computer System Implementation**]: Portable devices

General Terms

Design, Experimentation, Performance

Keywords

Heading direction; Force analysis; Mobile phones; Sensing; Compass correction; Localization; Activity recognition; Orientation; Magnetic field

MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA.

http://dx.doi.org/10.1145/2594368.2594392 .

1. INTRODUCTION

Despite tremendous amount of academic research, indoor localization is still struggling to come to mainstream. Based on numerous discussions with industries and start-ups, we are beginning to realize that much of the academic research has optimized for "location accuracy", while making simplifying assumptions about other aspects of the end-to-end system. Example assumptions include the availability of indoor maps, availability of dense WiFi infrastructure, orientation of the phone, known walking directions, etc. The consistent feedback we have received is that relaxing these assumptions is crucial at this point, perhaps even more than pushing the boundaries of location accuracy. Driven by this feedback, this paper concentrates on automatically sensing a user's global walking direction regardless of the orientation of the phone.

If solved well, a user's walking direction can offer benefits beyond inertial localization. Stitching crowd-sourced sensor data to infer detailed indoor maps is of increasing interest – the user's walking direction is a critical component in this. Elevator companies are envisioning that future elevators will be proactively fetched when the resident of a building begins to walk towards the elevator [2] – the walking direction is valuable here. Walking directions may also translate to the user's facing direction, enabling additional applications in augmented reality, social activities, and context-awareness. Of course, these benefits would be available only if walking direction is estimated through a generic stand-alone module, without relying on external information (such as maps, locations, WiFi access points). Moreover, to be able to tell the global walking direction (e.g., 45° North), the compass errors need to be mitigated. *WalkCompass* is focused on addressing these challenges.

It is natural to question the difficulty of solving this problem – one may ask *why not estimate the walking direction by projecting accelerometer signals to the horizontal plane?* While this is true in abstraction, the real situation is far more complicated when the phone is held in different orientations at various body positions. In pant pockets, for example, the phone swings along with the leg, continuously altering the phone's local coordinate system; free-flowing hands are similar. Users sway sideways in varying degrees while taking each step, which becomes pronounced when the phone is in the palm (e.g., when the user is checking emails). Inherent sensor noise further pollutes the signal. The precise walking direction is actually a micro-motion

that needs to be carefully identified, weighted, and averaged over at least two steps.

Even if the user's walking direction is successfully extracted in the phone's local coordinate system, translating it to global directions, say 45° North East, is an additional challenge. This is because the global (magnetic) coordinate system relies on the compass, which itself is susceptible to ferromagnetic interference in many indoor environments. All in all, the accelerometer signal that corresponds to the direction of walking gets blended into various other motion patterns that vary across users, environments, and phone models. Solving it generically, we believe, is non-trivial.

The core intuitions in WalkCompass are simple and can be explained in two parts. First, when a user walks, her heel strikes the ground creating a distinct vibration that resonates through the entire body. This vibration reflects on the accelerometer data across all holding positions, even when the user is holding the phone against her ears. WalkCompass uses this vibration as a reference, scans the signal backwards, and extract specific samples from a time window when the body's movement is dominantly in the heading direction. The signal is then processed with the gyroscope data to compensate for instability of the phone's coordinate system, such as when the entire phone is swinging in the pant pocket. This motion vector is then projected to the plane orthogonal to gravity, and averaged over few steps to converge upon the *local* walking direction.

The second problem pertains to translating the walking direction to the global magnetic coordinate system, but for this, the compass needs to be improved. WalkCompass's intuition is to treat the compass measurements as a signal, shaped by the earth's magnetic field, and by (the resultant of) other interferers in the ambiance. While separating this resultant interference is a difficult problem, the opportunity arises from walking. Since the phone moves in small steps, it observes staggered snapshots of the same interference – as if watching an object from different viewing angles – thereby enabling the possibility of triangulation. Of course, triangulation is possible only when the signal exhibits certain properties. WalkCompass exploits these properties to correct the compass direction in certain locations, and uses gyroscope based dead reckoning to track the walking direction between these locations.

Of course, the above is an over-simplification of our designed algorithm – several constraints need to be accounted for that are detailed in the paper. WalkCompass incorporates these algorithms into a functional system, with certain optimization for human variations and holding positions. Experiment results from 15 different settings show that the main limitation of WalkCompass is in its time to convergence. While an ideal system should be able to offer the direction in 2 walking steps, WalkCompass sometimes requires up to 5 steps. However, the direction accuracy is promising, with the 75^{th} percentile error being less than 12 degrees, in comparison to the compass which can be 26° even when held in the direction of walking. The accuracies scale across users, varying placements on the body, and while walking over different surfaces such as concrete, carpets, etc. Micro-benchmarks on magnetic interference calculation also exhibit promising results, demonstrating that the native compass errors can be appreciably reduced (when the user is walking). While there is still room for improvement, we believe WalkCompass can already be useful to other applications. We have demonstrated the system to a few companies – an YouTube video of the demonstration is posted here [3].

The main contributions of WalkCompass may be briefly summarized as follows:

1. *Analyzing the anatomy of walking patterns from the perspective of smartphone sensors.* This is performed through a synchronized analysis of a walking video and sensor readings, revealing meanings of each short segment of accelerometer readings. We believe our analysis tool could benefit other smartphone-based activity recognition systems.

2. *Coping with magnetic interference in indoor environments.* Designing various techniques to localize, quantify, and isolate the interference, with certain inspirations borrowed from noise cancellation techniques in wireless communication.

3. *Implementation of a functional prototype on multiple models of Android phones.* Experiment results from 15 different buildings indicating median error of 8° with around 5 steps to achieve convergence.

2. FEW NATURAL QUESTIONS

(1) **Despite substantial research on activity recognition and phone orientations, why is walking direction still an unsolved problem?**
While we have been somewhat surprised as well, our literature survey revealed some insights into this question. Most papers that have attempted this problem have done so in the context of a broader application, and have leveraged application-specific opportunities to resolve the challenges. For instance, [9] analyzes human motion in indoor settings and inherently requires a map of the place; authors leverage the same map to estimate the user's walking direction. Authors in [30] perform localization using information about the WiFi AP's location, and use the same AP locations to infer crude walking direction. [22, 19] assumes that the phone's initial orientation is known, and tracks the user's heading direction for a short time window after that. Zee [28] is one of the localization papers that addresses the problem without other assumptions, but admit that the solution is not stable and offers ambiguity – they mention that using maps can solve the problem at the expense of some latency.

A recent paper [5] attempts to solve the problem as a stand-alone module but uses highly controlled conditions for testing, namely, phone held in hand. MATLAB results are presented for two path traces from outdoor environments. Moreover, the solution is only in the local coordinate frame. Finally, several papers estimate the orientation of a smartphone on the human body. Phone orientation, while necessary, is insufficient for translating a user's walking direction to a global coordinate system; moreover, detecting the (local) walking direction is a separate problem. Knowing the orientation does not make this problem any easier.

(2) **Why not use the sequence of user's location to compute walking direction?**
In outdoor environments, this is true and trivial (assuming GPS energy consumption is not an issue). For indoors, however, walking direction is needed to compute location in the first place. This reliance is growing stronger due to two reasons. First, with more systems moving away from manual war-driving to crowd-sourced approaches, it is important to understand the motion patterns of the crowd, so that sensed data can be appropriately oriented. Second, pedestrian dead reckoning is becoming increasingly popular for localization due to its ability to track fine scale human movements (otherwise difficult with say WiFi or cell tower signals). As a result, WalkCompass cannot assume the availability of the user's location – the estimated walking direction will be needed to facilitate localization.

3. SYSTEM OVERVIEW

Figure 1 shows the key building blocks underlying WalkCompass, and their connections. We describe the flow of operation here, and expand on the technical details in the next sections.

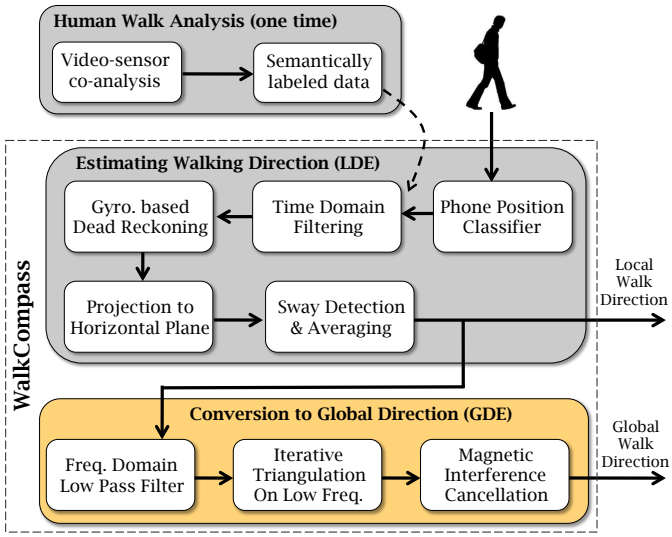

Figure 1: Flow of operations in WalkCompass. The input is the phone's sensor signals and the output is the local and global walking directions.

The overall WalkCompass system has 3 main modules, namely: (1) Human Walk Analysis (HWA), (2) Local Walk Direction Estimator (LDE), and (3) Global Walk Direction Estimator (GDE). HWA is a one-time analysis of human walking patterns that helps us understand how different forces act in different directions to keep the human body stable during a walk. We observe how these forces are manifested on the smartphone's accelerometer in different positions on the body, and use the insights to make key design decisions in LDE. Specifically, we identify the exact time-segment of the accelerometer signal that contains the forward motion of the body-part (e.g., leg, hand, chest) near which the phone is located.

The Local Walk Direction Estimator (LDE) receives the raw accelerometer reading from the smartphone, and first classifies the data to infer the phone's location on the body. The (time domain) accelerometer signal is then analyzed and the appropriate segment extracted based on HWA's recommendations. This segment contains the forces in the direction of the walk, but they are in 3D; moreover it is polluted by a constantly-rotating coordinate system (say when the upper leg swings while pivoted to the hips). The gyroscope is engaged to compensate for this rotation, and after some processing, the motion vector is projected to the horizontal plane, orthogonal to gravity.

Now, this vector is only for one step of one leg, and not necessarily the direction of the walk – humans extend their feet in side-ward directions, called sway, and the walking direction is an average of sway. The output from this averaging operation yields the walking direction in the phone's local coordinate system. Observe that this implies that with respect to the human's walking direction, the phone's orientation is now known.

The local walking direction may suffice for some applications; others need the global direction. In a perfect world, the global walking direction would be the angle subtended by the walking vector and the compass direction (i.e., North). For instance, if the walking vector points in the opposite direction of the phone's compass, then the global walking direction would be "South". However, the compass can be erroneous indoors due to ambient ferromagnetic interferences – the GDE module is tasked to compute the global walking direction despite these errors. To this end, the GDE module operates on the frequency domain, treating the sequence of compass readings as a digital signal.

As a person walks through indoor spaces, WalkCompass hypothesizes that the effect of nearby interferers would change quicker over time, and would be captured by the high frequency components in the compass signal. Filtering out the high frequency components, the low frequency components are expected to be from further away sources of interference, adding an offset to the compass reading. GDE employs an iterative algorithm that essentially attempts to localize this far-away interferer. The key intuition is to decompose the measured magnetic vector into the earth's magnetic vector (G) and the interference vector (I), and adjust the direction of G until all the I vectors intersect at the same point[1]. This value of G is inferred to be the actual earth's magnetic field. Of course, this triangulation may not be feasible in all locations (e.g., where many strong and opposing interfering sources are located far away). However, if some locations offer feasible results, the global walking direction can be computed there, and gyroscope-based tracking can be used at other (in-between) locations. The subsequent sections expand on each of the techniques, followed by micro-benchmark and full-scale system evaluation.

4. SYSTEM DESIGN

We describe the three modules, HWA, LDE, and GDE, for the case where the phone is carried in the pant pocket. This is actually the harder case, compared to the palm or in the

[1]Note that the magnitude of the earth's magnetic vector at a given location can be looked up in a database, so knowing the user's crude location, say the zip code, is adequate.

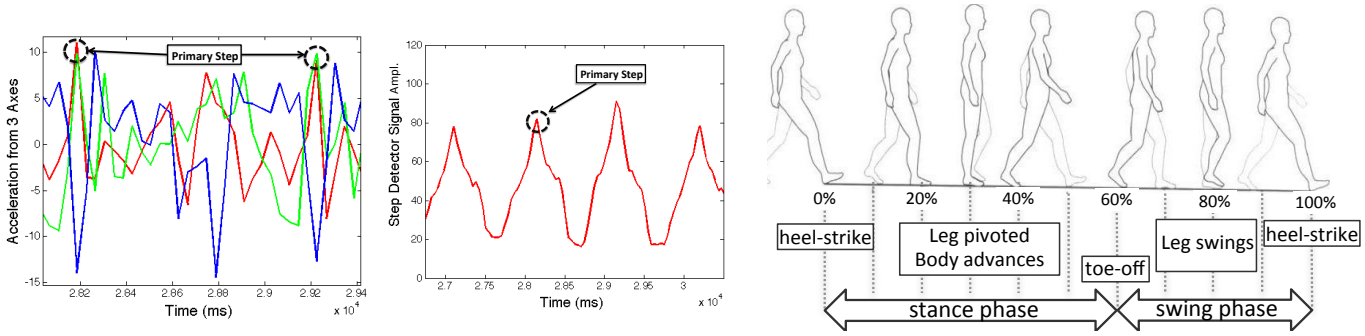

Figure 3: Accelerometer data from human walking patterns with the smartphone placed in the pocket: (a) Correlated jerks on all 3 axes on the accelerometer indicating a heel-strike on the ground. (b) The peak detection algorithm detects the timing of the heel-strikes. (c) The gait cycle of human walk composed of 2 phases: the first 60% between two heel strikes is called the *stance*, and the next 40% is called the *swing*.

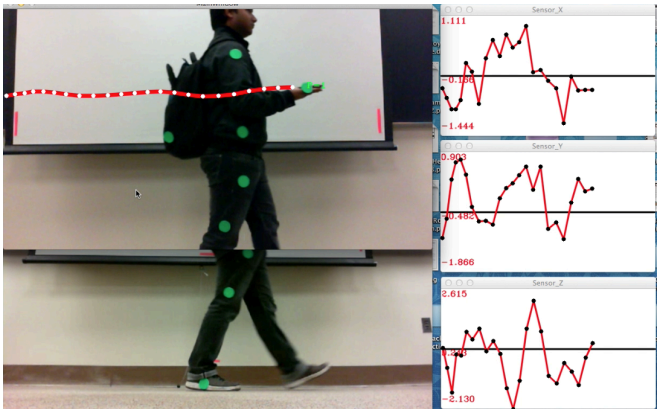

Figure 2: Matching video and accelerometer for labeling measured readings – the green circles are tracked across video frames to compute their respective motion vectors, and later matched against the vectors derived from the accelerometer.

swinging hand. At the end of this section, we will discuss how the techniques can be generalized to other placements with small modifications.

4.1 Human Walk Analysis (HWA)

Our goal in HWA is to be able to label the different parts of an accelerometer reading with the corresponding micro-actions that the user is performing at each time point. The hope is that this analysis would reveal at least one segment of the signal that is precisely along the walking direction, and more importantly, is least polluted by other micro-actions of the limbs. Of course, numerous studies, especially in the domains of physical therapy and computer animation, have analyzed the acceleration of various body parts during the walk cycle [13, 14, 12]. Our study, however, is from the view of the smartphone sensor, placed at different parts of the body, inside different outfits.

We set up a system in which 2 video cameras were placed focusing various body parts of a walking person, say John. John carried phones at multiple positions in the body – at each of these locations, we attached a green marker to help track the motion of the phone through the video frames. All phones and the video cameras were time synchronized (to sub-milliseconds) by shining a bright light on all their cameras simultaneously. From the gathered videos, we tracked each green marker over time and translated them into motion in the X, Y, and Z axes of the phone. We matched the visually-computed motion with the accelerometer signal to synchronize them as well as possible. This process helps in matching the different ranges of motion (one in pixel space and the other in sensor signal space), as well as for synchronizing timing shifts due to fluctuations in accelerometer sampling (details omitted in the interest of space). Figure 2 shows the outcome – we are now able to click on any time window on the accelerometer signal and observe the corresponding video clip.

We distill 2 useful findings from this experiment that we use later to design the system. Here the limb that carries the phone is referred to as the *primary* limb, and the other is called *secondary*.

(1) Heel Strike is an Anchor Point

The maximum amplitude and fluctuation in the acceleration, along all the 3 axes, occurs when the heel of the human strikes the ground. This holds true across all positions on the body, including hand, backpack, pant pocket, shirt pocket, etc. – only the magnitude of the jerk is different in each of them. This general property serves as a valuable signal marker – an anchor point – from which other stages of the walk can be analyzed. To isolate this heel-strike with high reliability, we add all the 3 dimensions of the accelerometer signal, and take the signal envelope of this summation, and pass through a simple peak detection algorithm (Figure 3a and 3b). The outputs of the algorithm are the precise timings of the peaks.

(2) Swing Before Heel Strike

Between consecutive heel strikes, the legs go through two main phases – a "stance" and a "swing" as shown in Figure 3(c). Stance is the phase when the upper body and leg move forward while pivoted to the feet on the ground; swing is the phase when the leg swings forward, pivoted to the hip, to overtake the upper body. During the swing, the acceleration is positive in the first half of the swing, and negative in the second half (to bring the leg to a stop for a heel-strike).

Further, say the swing lasts from time t_1 to t_9 (Figure 4) – the acceleration is maximum at time t_3, then its zero at time t_5, and then the deceleration is maximum at time t_7.

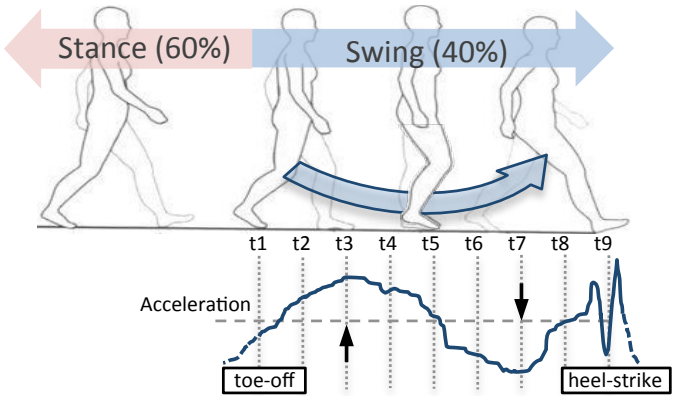

Figure 4: The acceleration at various points of the swing phase during the human walk cycle.

Our first approach was to extract the accelerometer data around t_3 and t_7, reverse the sign for the data at t_7, and average them to derive the user's walking direction. This should cope well with accelerometer noise since hardware noise is Gaussian with zero mean, hence should cancel well upon averaging. However, we observed that the data around t_3 is polluted by the motions of the secondary leg and the upper body. On the other hand, these secondary and upper body motions are much less prevalent around time t_7 – perhaps because the body is trying to stabilize – making the signal pristine. Hence, the Human Walk Analysis (HWA) module prescribes the region around this t_7 time point as the signal containing the walking direction. Of course, this is an approximation, since the above analysis assumes that the swing is occurring uniformly in time. To be accurate, we cannot assume that the mid point of the swing is at t_5, and the deceleration is maximum mid-way between t_5 and t_9. The subsequent discussions in LDE will account for these factors.

4.2 Local Walking Direction Estimator (LDE)

The LDE module receives a user's accelerometer readings as an input and runs the raw data through a position classifier to detect where the phone is located on the body. Past work has reliably solved this problem [29, 27] – we have borrowed these solutions, and depending on the placement, apply minor variants of LDE. For ease of explanation, we describe LDE entirely for the case of pant pockets, and discuss the variants at the end of the section.

Filtering: Assuming that the phone is in the pocket, the data is then received by the Time Domain Filtering module. This module computes the heal-strike peaks in the signal, looks back from the peak to extract the segment with maximum deceleration in the second half of the swing (as described earlier in HWA). More precisely, the start of the second half needs to be estimated first, for which WalkCompass looks for the change in the sign of acceleration from positive to negative. Denote this time t_i and the time of next peak, t_j. WalkCompass extracts the time point of maximum deceleration within $[t_i,\ t_j]$ – denote this as time t_x. To cope with accelerometer noise, typically Gaussian with zero mean, WalkCompass extracts a short segment around t_x[2]. WalkCompass passes this segment through a low pass filter to remove high frequency noise and potentially other signal pollutants, and forwards the samples to the gyroscope based dead reckoning module.

Dead Reckoning: The dead reckoning module fetches the gyroscope readings at these exact time points (corresponding to these signal samples) and observes the angular rotation of the phone. By rotating back the phone's coordinate system to a reference time point (i.e., rotating the phone in the reverse direction of the gyroscope readings), the samples are now brought to a stable coordinate system. The samples are now averaged, and the sign reversed. This is the walking vector in 3D space.

Projection to Horizontal: The walking direction in 3D space is then projected to the plane orthogonal to gravity, also called the *walk plane*. The gravity vector is of course not obvious from the raw accelerometer reading – we estimate the gravity vector by fusing the accelerometer and gyroscope data, and applying the complementary filter [1]. WalkCompass projects the 3D walking direction into the plane perpendicular to this computed gravity vector.

Sway Detection: In bipedal movement, each leg moves the body in a slightly lateral direction – the walking direction is actually a resultant of these two lateral motions. Thus, the walking direction computed so far is slightly offset from the actual walking direction. Further, the body also sways during the walk, introducing additional forces unaligned to the walking direction. Fortunately, all these movements are reasonably symmetric with respect to the heading direction in the horizontal plane. Therefore, the resultant of the primary and secondary movements mitigates these lateral movements and sway-induced errors. WalkCompass estimates the walking direction on the secondary leg using the same technique as the primary, except that it uses a second smaller peak in the accelerometer signal. These smaller peaks are indicative of when the secondary heel strikes the ground. The final resultant vector is passed through a median filter to remove random jerks on the phone during the walk. The output of the median filter is forwarded as the final output of LDE – this is WalkCompass's estimate of the walking direction in the phone's local coordinate system.

4.2.1 Coping with Phone Placement

Past work has developed classifiers that reliably discriminate the phone's location on the body (palm, pant pocket, swinging hands, etc.). The key technique is to observe the energy density from acceleration, as well as the magnitude of peaks during the walk. WalkCompass borrows this classifier with a few minor adjustments – this is not our contribution.

Once the phone's location is known, we apply a few minor modifications for the case when the phone is held in the hand (both palm and swing). Specifically, instead of summing the (x, y, z) accelerometer signals, we first compute the variation along the gravity dimension (which is

[2]Taking a sample only at t_x is susceptible to noise, instead averaging over a few samples is expected to dampen the noise.

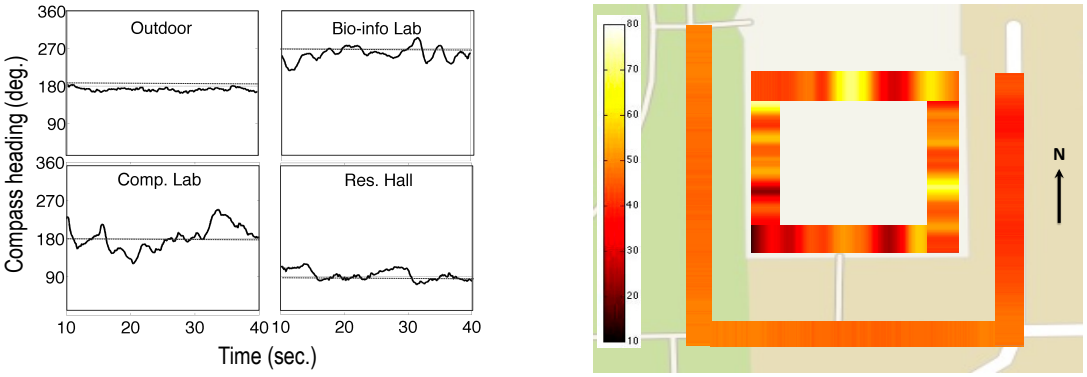

Figure 5: (a) Magnetic fluctuations in 3 indoor environments far higher than outdoors. (b) Heat map of magnetic intensity across the entire corridor of a science laboratory. WalkCompass performs evaluation in these settings.

not necessarily the z axis). The intuition is that the phone bounces maximally along the gravity, and the HWA module shows that the bounce-peaks are well correlated to the heel strikes. Thus, once we anchor the accelerometer signal to these bounces (which are symmetric for both legs), we apply the same segment extraction scheme. Fortunately, the time point when the leg's deceleration is maximum, is also the time when the upper body is getting stabilized from full-fledged forward motion. Selecting this time window is effective for hand-held phones. The case for shirt pockets is slightly more difficult, since the bounce is softer. Nonetheless, the bounce is still visible among the experiments we performed (but may be less reliable for a soft walker, or a child). We leave jacket pockets and bags to future work.

4.3 Global Direction Estimator (GDE)

Translating LDE's output to the global coordinate system entails 2 steps: (1) finding the true magnetic north in the phone's coordinate system, and (2) measuring the angle between the true magnetic north and LDE's output. Step 2 is trivial, and in an ideal world, even step 1 is simply the direction given by the smartphone's compass reading. In reality, however, the smartphone compass is heavily influenced by magnetic interferers in the surroundings, especially in indoor environments. Figure 5(a) shows the compass readings from Android and iPhones in 4 different environments – an outdoor sidewalk, a bio-engineering building, a computer science building and a residential hall. In 3 of these 4 cases, the compass deviates heavily from the actual walking direction. Figure 5(b) shows a more detailed view – a heat map of the magnetic intensity when the user walks with the phone in a rectangular corridor. The intensity varies frequently and ranges from 10 to 80 micro-tesla, indicating high magnetic interference. As a result more than 95% of the corridor incurs errors greater than 23 degree offset. However, on the outdoor path around the building, the magnetic intensity remains close to its expected value, around 53 micro-tesla at that location. In light of this, GDE concentrates on improving the compass, which is likely to benefit other applications that utilize the native compass app.

Background on the Magnetic Compass

The Earth behaves like a giant magnet surrounded by lines of magnetic flux [24]. A freely suspended magnetic needle aligns itself with these lines and directs itself towards the magnetic North of the Earth. Although these magnetic flux lines converge at magnetic poles, they appear parallel in a given (small) region, due to Earth's large diameter. Thus, when we walk with a smartphone in outdoor settings with zero magnetic influence, we observe that the compass readings – a 3D vector – are parallel to each other. However, as we introduced an artificial interference near the phone[3], we observed that the measured compass vectors were distorted. The distortions were in the direction of the resultant of the Earth's magnetic field (G) and the field caused by the interferer (I) – shown in Figure 6(a). Now, with multiple interferers close to the phone, the measured compass readings changed quickly since small displacements brought the phone relatively closer to some interferers and further from others. When the interferers were moved far away, the magnitude of the interference reduced, and the fluctuations in the readings naturally subsided. The measured compass vector exhibited an offset from the actual magnetic north, and this offset changed slowly over time (Figure 6(b)).

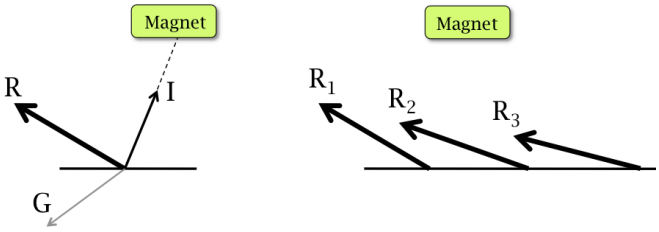

Figure 6: (a) Measured compass vector (R) is the resultant of earth's magnetic field (G) and the interference vector (I). (b) Compass vector R changing slowly as user moves.

Intuition for Compass Correction

The above observations, although unsurprising, offered us an intuition. We recognized that when the compass is moving, the data it records is essentially multiple snapshots of the magnetic interference from slightly changing angles. Assuming the interferer is far away and stationary, the consecutive compass vectors should make small angles between each other. Now, if the earth's magnetic north vector was

[3]Interferers can be permanent magnets, called *Hard Iron* interference, or any ferromagnetic material, called *Soft Iron* interference.

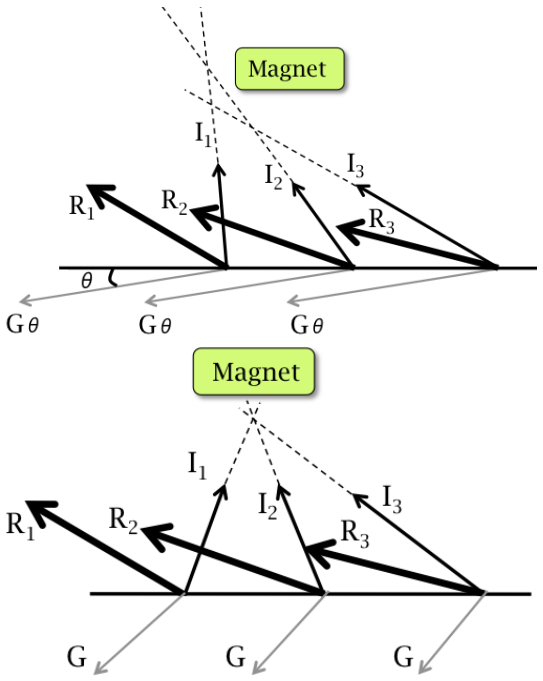

Figure 7: (a) With incorrect G vectors, the I vectors do not coincide. (b) Upon iterating over θ, for some value of G_θ, the vectors coincide.

indeed known, then subtracting this vector from the compass vectors, should have offered the interference vectors, and more importantly, these interference vectors should have intersected at the location of the interferer. In reality, since the earth's magnetic north is not known, we asked: *what if we iterate over all possible vectors of magnetic north until we find the interference vectors intersecting at one location.*

Figure 7 illustrates the idea graphically. Compass readings [R_1, R_2, R_3...] correspond to a moving user recording the readings at times [t_1, t_2, t_3...]. Now, since the direction of G is not know, it is possible to assume an arbitrary direction, say G_θ, and subtract from the R vectors, resulting in interference vectors, I_1, I_2, and I_3. If G_θ is incorrect, the interference vectors are not expected to intersect at a common point. However, when iterated for all values of θ, the correct value of G may be expected to offer a common point of intersection, shown in Figure 7(b). If this intersection indeed occurs, then we can select the corresponding G_θ, and use that as an estimate of the true magnetic north.

A natural question is: *does the IMT algorithm assume only one interferer in the ambiance?* While it may appear to be so, observe that the interference vector we estimate could actually be the (vectorial) sum of all interferences in the ambiance. This is modeled under the principle that different force vectors can be linearly added and represented by a single resultant vector. Of course, there is no reason to believe that these resultant vectors would originate or culminate at a single source (the point of intersection). However, given that the spatial gaps between R_1, R_2, R_3 is small (say, three consecutive samples at 25 Hz), the resultant vectors are likely to be offset by small amounts, and hence, could be expected to meet at nearby points (a heuristic). The rest of this section details this algorithm, called *Iterative Magnetic Triangulation* (IMT). Several pre-processing steps and fine-tuning are necessary to make the algorithm generic and robust to widely varying real-world conditions.

Iterative Magnetic Triangulation (IMT)

We describe 5 steps of the IMT algorithm next.

• Step 1: Vector Selection
The compass continuously provides a series of R_i vectors, however, not all these vectors are suitable for the IMT algorithm. For example, when the user is not moving, the compass reports unchanging magnetic vectors that are parallel to each other. The interference vectors for those samples will also be parallel, rendering triangulation infeasible. On the other hand, if the phone sways significantly (on the axis perpendicular to the of walking direction), then it breaks IMT's model that the phone is dominantly moving along the walking direction. Finally, the triangulation heuristic requires the interferers to be sufficiently far, so that within a small time window, the magnitudes of the interference vectors are almost equal. Recall that with nearby interferers, the magnitudes change much quicker.

In view of these constraints, IMT treats the compass data as a signal and first passes it through a low pass filter. This eliminates the high frequency components corresponding to nearby interferers, and leaves the influences of far away and strong interferers. On this residue signal, IMT opportunistically selects 3 consecutive vectors that are not parallel, and whose projections on the sway-axis have negligible variation. We find reasonable number of vector triplets $< R_i,\ R_{i+1},\ R_{i+2} >$ that satisfy this criteria. Recall that even if WalkCompass obtains the global walking direction in a few spots, it can use the gyroscope to track the user in between those spots (since the gyroscope is not affected by magnetic interference).

• Step 2: Iteration and Triangulation
IMT has a reasonable (though not precise) estimate of the magnitude of G, based on the phone's crude location (at the granularity of, say, zip codes). This can be found from the International Geomagnetic Reference Field database [6]. However, the direction of G is unknown and IMT iterates over all possible values of θ. Thus, for a given θ, IMT subtracts G_θ from each of $< R_i,\ R_{i+1},\ R_{i+2} >$ to compute $< I_i,\ I_{i+1},\ I_{i+2} >$, and then observes how the interference vectors intersect. When the intersection points are tightly clustered (defined in more detail later), the corresponding G_θ is selected as the earth's magnetic field. If multiple values of θ present tight clusters, IMT chooses the values of θ that is closest to the compass reading. Finally, if no tight clusters are found, IMT attempts the same operation on the next valid vector triplet, or waits a few seconds to get fresh data from the walking user.

• Step 3: Refining Magnitude of G
Unfortunately, we observed that in some cases, the rough magnitude of G (looked up from the Reference Field database) provides unreliable results. This is due to inaccurate locations as well as unknown heights (i.e., a smartphone on the 5th floor of a building may observe different geomagnetic forces compared to sea level values from the

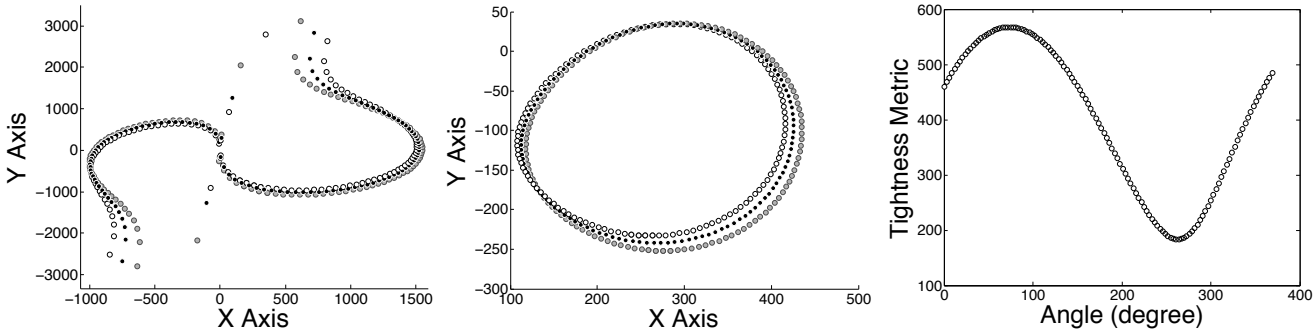

Figure 8: The locus of two interference vectors when iterating over geomagnetic vectors. If the magnitude of the geomagnetic vector is incorrect then the loci creates (a) an irregular shape, otherwise, (b) the loci creates a regular ellipse/circle shape. (c) Varying values of tightness with increasing θ – IMT picks G_θ corresponding to minimum tightness at around 260.

database). Importantly, we find that when the magnitude of G is erroneous, the locus of the intersection points of $< I_i,\ I_{i+1},\ I_{i+2} >$, computed from all values of θ, follow erratic shapes (Figure 8(a)). On the other hand, when the magnitude of G is close to the correct value, this locus forms a reasonably closed-formed shape, resembling a circle or an ellipse (Figure 8(b)). In view of this, IMT checks the shape of the locus and iteratively reduces the magnitude of G until the locus forms the expected shape.

• Step 4: Picking Tight Intersection Clusters
Ideally, each pair in the inference vector triplet should intersect at the same point. In practice, however, the points of intersection form a cluster. IMT picks the value of θ for which it finds the tightest cluster of intersections. We define the "tightness" of a cluster as the sum of all pairs of points in that cluster.

• Step 5: Final Global Walking Direction
Figure 8(c) shows the variation of *tightness* values for a full iteration of G_θ for a given vector triplet. IMT finds the lowest point of this graph (i.e., the minimum value of tightness), and the corresponding G_θ (263° in this example) is announced as the estimated geomagnetic north. Note that this can also be used as the new compass output, improving the inherent quality of the compass. Finally, WalkCompass compares the local walking direction against this estimated north, and outputs the global walking direction of the user. Algorithm 1 presents the pseudo code for the IMT algorithm.

Algorithm 1: Pseudocode for the IMT algorithm

```
Data: continuous magnetometer data
Result: θ_est
< R_i, R_{i+1}, R_{i+2} > = Vector Selection(filtered input);
r_G = geomagnetic intensity;
while r_G > threshold do
    for θ_G ← 0∘ to 360∘ do
        G_θ = vector with magnitude r_G and angle θ_G ;
        I_i = R_i − G_θ ;
        I_{i+1} = R_{i+1} − G_θ ;
        I_{i+2} = R_{i+2} − G_θ ;
        < P_i, P_{i+1}, P_{i+2} > = pairwise intersections of
        < I_i, I_{i+1}, I_{i+2} > ;
        t_θ = sum of all pairwise distances between P_i,
        P_{i+1} and P_{i+2}. ;
        Store < (t_θ, θ_G) > in T ;
        if Locus of < P_i, P_{i+1}, P_{i+2} > goes irregular
        then
            reduce r_G exponentially ;
            break ;
        end
    end
    θ_est = θ_G for which t_θ is minimum in T ;
    return θ_est ;
end
```

5. EVALUATION

5.1 Implementation and Methodology

WalkCompass has been implemented on Android, using the Jellybeans version, and tested using a variety of Samsung phones. The codebase has also been replicated on MATLAB to test optimization modules offline – the appropriate optimization have are ported back into the phone. Figure 9 shows a screenshot with the gray cone(thicker) showing the compass direction; the green cone(thinner) denotes the user's walking direction. The width of the cone is proportional to the variance of walking direction – a useful visualization for debugging in real conditions.

WalkCompass experiments were performed with 6 users who volunteered to walk with the phone in different environments, using varying holding positions. The environments were mostly UIUC's science and engineering buildings with heavy magnetic influence – this subject's WalkCompass to a stronger test. For each test, we asked the users to walk along established corridors – this is because we used the Google satellite view to compute the global truth in walking directions. Some experiments were also performed in Wal-mart, houses, and apartments, and the global truths were computed similarly. We also walked along with the users noting down when they changed the orientation of the phone, or how they placed the phones in the pocket. We report results against the baseline of the phone's compass.

5.2 Performance Results

We intend to concentrate on the following questions:

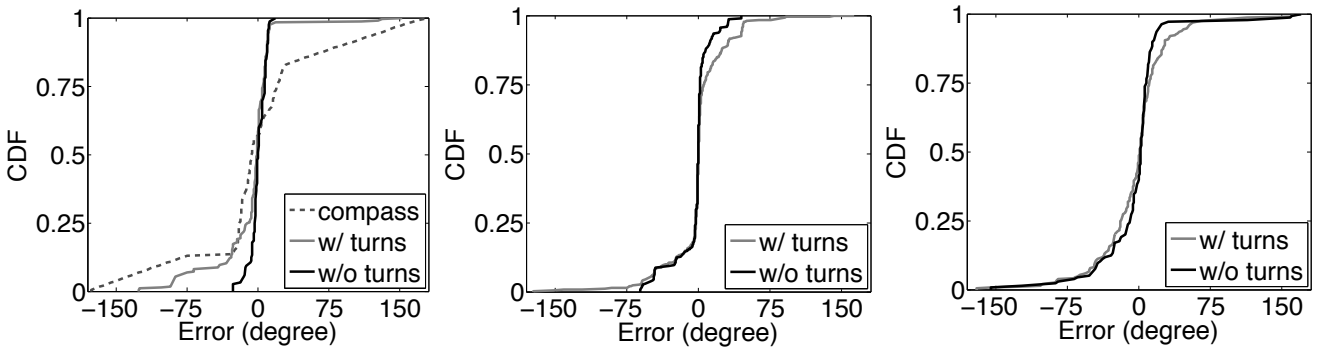

Figure 10: CDF of WalkCompass's LDE error with and without turns. (a) Phone held in the palm. (b) Phone carried in the pant pocket. (c) Phone held in swinging arms.

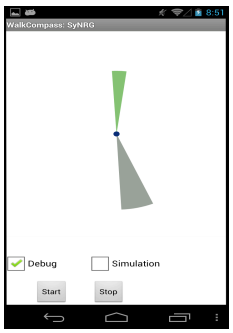

Figure 9: (a) WalkCompass screenshot. (b) QR code for WalkCompass demo.

- The accuracy of recognizing the walking direction in the phone's local coordinate system (i.e., results from outdoors and some interference-free indoor environments) – Figures 10, and 11. Speed of change detection, especially after sharp turns (Figure 12).
- Coping with different holding positions (Figure 13), and different orientations such as portraits to landscapes (Figure 14). Robustness when user is bear foot or wearing shoes, and walking on various surfaces, such as carpet, concrete, etc. (Figure 15).
- Accuracy of compass error correction with IMT – Figures 17, and 18. Its performance under various regimes of magnetic interference (in labs,office,house, and outdoor settings) – Figures 19, and 20.

Accuracy of Local Walk Direction (LDE)

Figure 10 plots the CDF of local walking direction error, estimated from all traces, across all users, and across all environments. Figure 10(a) plots the case when the phone is held in the palm (in the browsing position) – some are in portrait, some landscape, and some users held the phone at an angle to the walking direction. Figure 10(b) and (c) plot the case of pant pocket and swinging hands, respectively. WalkCompass's error is computed as the difference with the true walking direction (computed manually for every trace from Google's satellite view). Since we are unsure about the global truth near the turning positions (recall that users may be making soft turns), we show the error distributions including and excluding the turns. While excluding turns, we have removed 6 steps at the each turn.

We could not develop a meaningful baseline scheme for comparison, hence, we use the following. For the palm position in Figure 10(a), we record the orientation of the phone only when it is pointed in the direction of walking. We smoothen this data through a low pass filter (as a way of canceling out periodic perturbations) and plot its error CDF – denoted "compass". Evidently, even when the phone is pointing in the forward direction, and the data smoothened, the fluctuation is appreciable. Since we could not learn the phone's orientation in Figure 10(b) and (c), we do not plot the compass data in these two graphs.

Evident from the graph, WalkCompass exhibits low median error for a sizable fraction of the scenarios, and the errors are well distributed on both sides of the walking direction. This suggests that the heel-strike jerk manifests itself prominently across all these phone postures. The performance degrades when the hand swings – in fact, performance for one of the 5 users is poor, significantly skewing the distribution from the center. On close observation, we noticed that this user's hand-swings are embedded with several rotational motions of the wrist, and even though they are repetitive, WalkCompass is unable to cancel them out. For all other cases, the performance is reasonably consistent. For the palm, the median error is at $\pm 5^\circ$, compared to $\pm 23^\circ$ for the phone's native compass app (with low pass filtering). For the pant pocket, the median is at $\pm 3^\circ$, and for the swinging-hand, $\pm 8^\circ$.

Figure 11 breaks down the results for each individual user, demonstrating that the LDE module is fairly robust to different walking patterns. In all these traces, the user walked for around 20 steps on average before taking a turn, and has taken around 25 turns in total. We show both the median and the 75^{th} percentile to reflect the robustness of the system. When carried in the palm or pant pocket, the errors are around 6°, except for one user who experiences around 20° error. While this is not ideal, and leaves room for improvement, we believe it is still useful for various applications. An elevator company keen on automatically dispatching elevators based on approaching users finds such error margins tolerable.

Direction Change Detection

Figure 12 zooms into the turning behavior of users – this is a representative graph that plots the error when a user makes

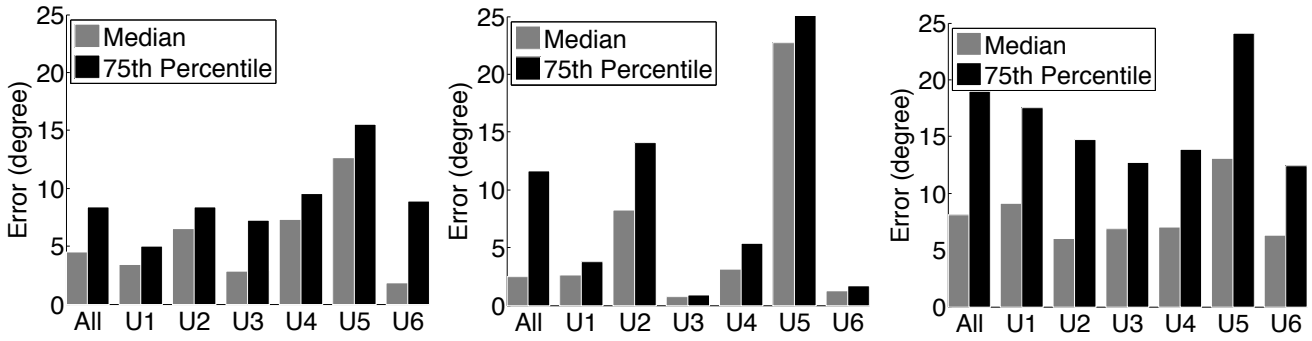

Figure 11: The median and 75^{th} percentile of error per user without turns. Phone carried in (a) palm, (b) pocket, (c) swinging-hand.

a 180° turn. This is the worst case behavior, and the graph shows the time it takes to converge to the actual walking direction. Evidently, the error decreases at a steady pace with more number of steps after the turn, and converges at around 5 or 6 steps. We believe these results are slightly conservative since the ground truth is assumed to be the intended direction in which the user is walking, and does not account for how the user actually walked. WalkCompass measures each of the micro-deviations the user makes at each step, and is hence slightly penalized here.

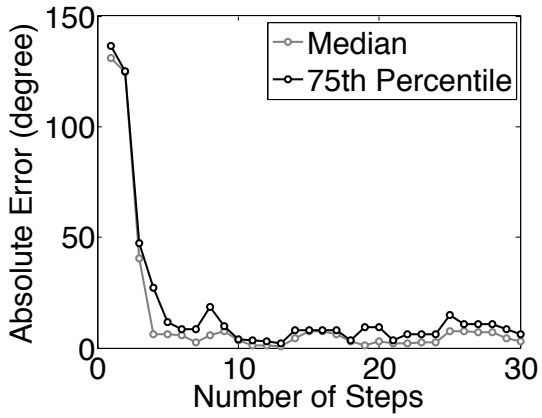

Figure 12: Error convergence after sharp turns. Both median and 75^{th} percentile error converge after around 4 or 5 steps.

Toying with the Phone

Users orient their phones in various ways, perhaps changing from a portrait to landscape for a video, raising the phone for a phone call, or just tilting to check an email. Figure 13(a) shows the changes in the phone's compass directions, while WalkCompass's estimated direction continues to point in the user's heading direction. Since LDE computes the force in 3D, and takes the projection on to the horizontal plane, the walking direction is estimated for all orientations (even when the user is holding the phone against the ears for a call). Figure 13(b) plots the distribution of errors across all users and traces.

Figure 14(a) zooms in to the same results and shows a scatter plot from many users toying with the phone. For easier visualization, all the traces have been offset to a common walking direction of 250°. Evidently, LDE copes consistently

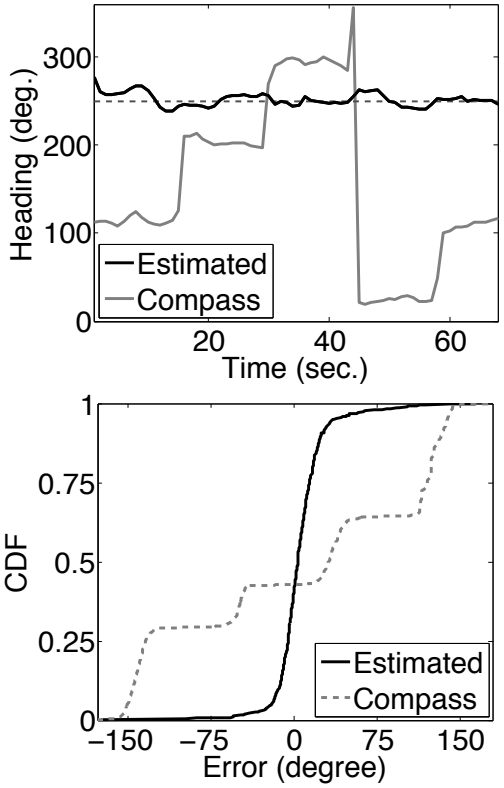

Figure 13: (a) Examples of traces in which users frequently changes the orientation of the phone. (b) CDF of error across all traces.

well with toying of phones and tracks the walking direction correctly. Some points fall far from the 250° line, however, observe that not too many points are consecutive, indicating that the large errors do not persist for long. To provide a sense of how much the phone was toyed with, Figure 14(b) shows the angles to which the phone was pointing across all the experiments.

Impact of Walking on Different Surfaces

Given that the success of LDE relies on correctly detecting the accelerometer signatures during a walk, we evaluate walking on various surfaces, with and without shoes. Users in our experiments walked barefoot and with shoes, on different surfaces, namely, carpet, and tiled floors. Figure 15

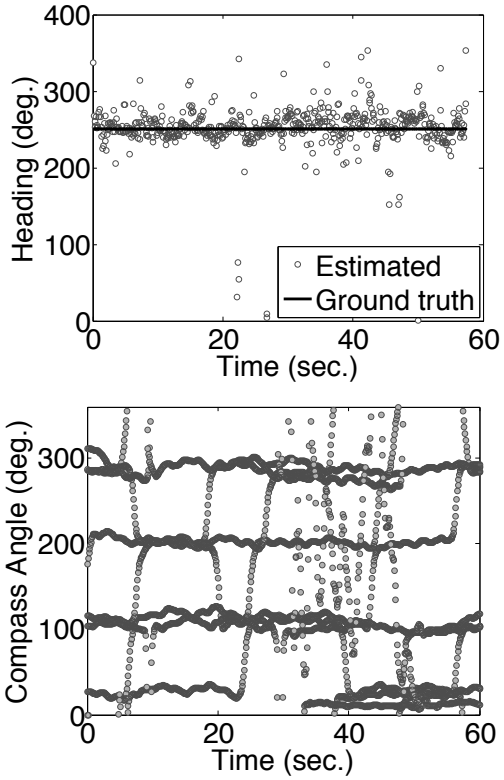

Figure 14: (a) Estimated heading for multiple users when they randomly change the phone's orientation. (b) Tracking the angle in which the phone is pointing to understand the extent of toying.

shows the median and 75^{th} percentile errors for each of these [footwear, surface] tuples. Observe that the errors are not affected by any specific surface – the heel strike is reasonably robust and lends itself across all these scenarios.

Staircase and Backward Walk

We evaluate different modalities of walking, such as walking up and down staircases, and walking backwards. Figure 16(a) plots the CDF for staircases while Figure 16(b) shows the performance when the user walks backwards. For both scenarios, the phone was carried in different orientations and positions in the body. As with regular walking, the performance of a specific individual was relatively worst than others, especially for the hand-swing. Otherwise, the performance was stable. Finally, we note that in all cases, the error of walking did not accumulate over time; even when users walked for long duration and experienced some error due to some jerks on the phones or other actions, a new heel-strike would reset the errors. We believe this is a desirable resilience property in any sensing system, and WalkCompass possesses it. We have omitted these results in the interest of space.

Global Walking Direction Estimation (GDE)

This section focuses on evaluating the accuracy of the global walking direction estimator (GDE) module, and specifically the *Iterative Magnetic Triangulation* (IMT) algorithm. To test this algorithm in diverse conditions, we select 15 different locations from various regimes, including science and

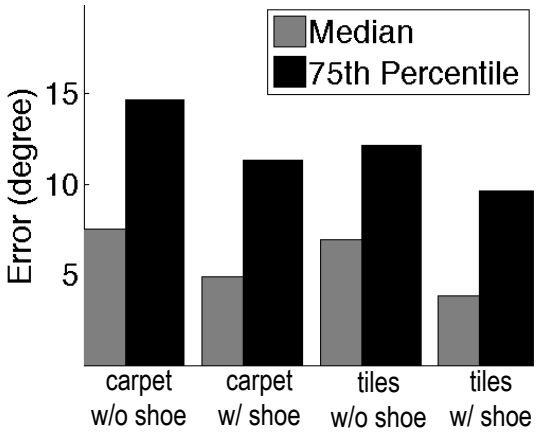

Figure 15: Walking on different surfaces (carpet and tiled floors) with and without shoes.

engineering laboratories, social places, houses, libraries, and outdoors. We apply the IMT algorithm on the magnetometer data and compare the results with the smartphone's native compass app. Since IMT's task is to only mitigate the magnetic interference (and not estimate walking directions), we hold the phone in the palm with the front of the phone facing the walking direction. We carefully record the ground truth from Google Satellite view.

5.2.1 Error distribution

Figure 17 plots the angle histogram across all traces, comparing IMT and the phone's compass against ground truth. Ideally this plot should have a spike only at 0°, which is the ground truth. Evidently, the compass measurements scatters around 60° on both sides of 0°, occasionally reaching up to 90°. The long histogram bar around 25° suggest that that's the common case error. IMT, in contrast, produces errors of 10° from the ground truth. Figure 18 shows the cumulative distribution of absolute error for both IMT and compass. The curve rises steeply for IMT and its value remains less than 15° for 75% of the traces, whereas the raw compass is at 37° for the same 75^{th} percentile. The median error of IMT is around 7°, less than a third of compass error. However, for very high magnetic interference, where the compass error is more than 50°, the performance of IMT degrades, although still remaining better than the compass. Figure 19 compares the median error of IMT and native compass for each of the 15 places. IMT is consistently half or one-third of the compass. This is encouraging since even outside WalkCompass, the techniques from this paper can be useful in improving the compass app in phones, especially in magnetic rich places where they are needed most.

5.2.2 Distribution of interval between effective triplets

Recall that IMT estimates the true compass direction on a set of three magnetic vectors, called a *triplet*. Ideally, many triplets should be available, so that WalkCompass can frequently estimate the true North. It is also advantageous to have the triplets evenly spread over the path of a walk – this allows better interpolation (using the gyroscope) between these correct points. Figure 20(a) shows the distribution of the intervals, in seconds, between two consecutive triplets found during the experiments. Although triplets tend to be collocated, we have found triplets separated by around 7 sec-

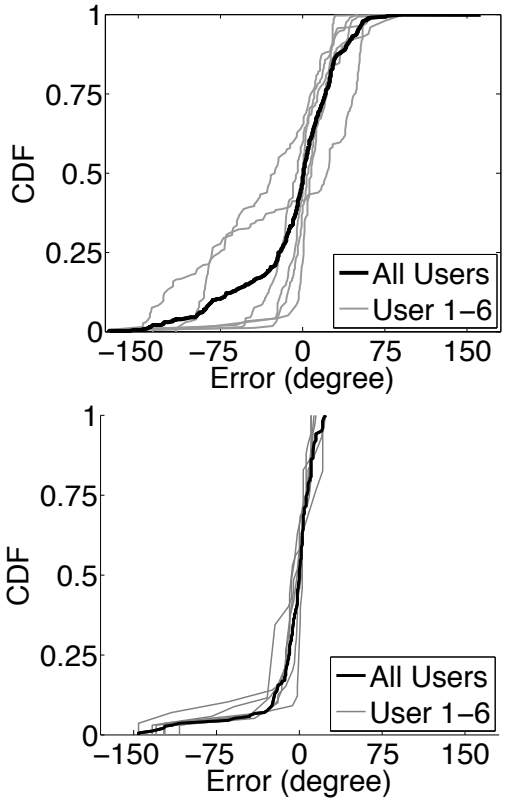

Figure 16: (a) The cumulative error distribution for walking up and down staircases. (b) CDF of error for walking backwards.

onds (i.e. every 7 steps) even at the 90^{th} percentile. Figure 20(b) shows the number of triplets found in each individual experiment. The density is sufficient to continuously and reliably track the true geomagnetic North during the walk.

6. LIMITATIONS AND DISCUSSION

We discuss a few limitations and opportunities.

• **Tail of Error Distribution:** While median and 75th percentile of accuracy is quite reliable with WalkCompass, in certain settings the performance drops sharply. This is particularly evident in areas where the compass error is extremely high – greater than 45° – and in other cases where certain users vary the phone's orientation while swinging their arms. We do not have a strong handle on why these

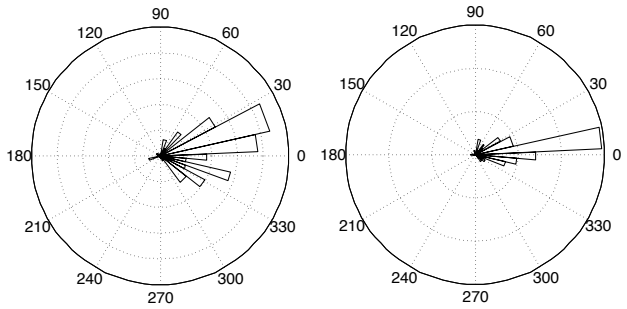

Figure 17: The angle histogram of (a) phone's native compass app, (b) IMT's estimates.

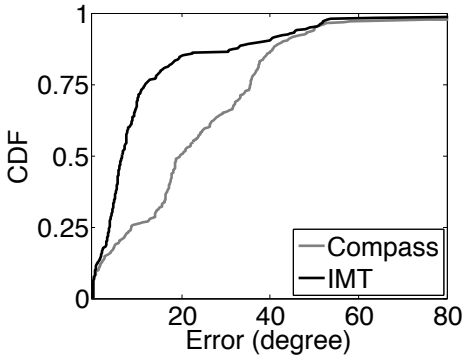

Figure 18: Comparing CDF of absolute error between the compass and IMT.

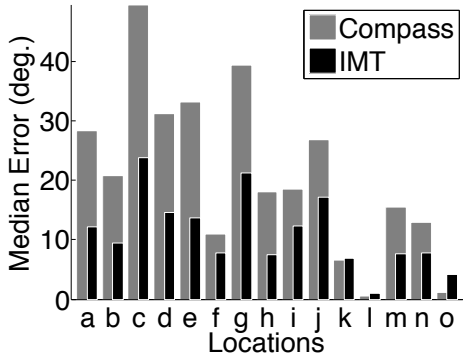

Figure 19: The median error comparison between IMT and compass across all 15 places.

cases occur – deeper engineering and fine tuning is necessary to cut-back on the tail of the error distribution.

• **Need to Walk a Few Steps:** The proposed IMT algorithm cannot use any magnetometer data to infer the magnetic north – only certain magnetometer snippets produce the correct answer. This indicates that the user would have to walk a distance before WalkCompass can estimate the magnetic North. While this may be tolerable for human walking applications, additional research is needed if the phone's native compass has to be improved. Our ongoing work is investigating methods to infer the true North from any snippet of magnetometer data.

• **Beyond Walking:** While WalkCompass has not been tested for other forms of human locomotion, such as wheel chairs, skate-boards, biking, etc., we believe the core techniques may still apply so long as there is a repetitive force in the heading direction. Pushing the wheel for wheel chairs, swinging the leg for skateboards, and rotation of the legs during cycling, all seem to offer this opportunity. We plan to test this in future.

• **3D WalkCompass:** This paper investigates human walking direction on 2D, but we believe that the core techniques can be scaled to 3D, albeit some additional complexity in the IMT algorithm. We leave this to future work.

7. RELATED WORK

We discuss past work on detecting walking directions.

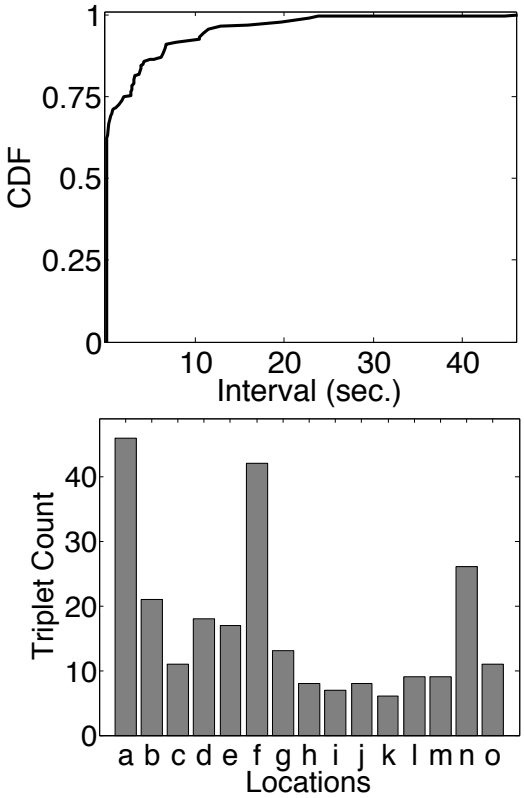

Figure 20: The distribution of the gap between two consecutive triplets(a) and number of triplets found per location(b).

Finding Heading Direction

Body-pasted sensors: Early work from anatomy analysis, skeletal parsing, and computer animation have studied the problem of walking direction using body-pasted sensors [16, 32, 17]. Smartphone sensors are characterized with far more degrees of freedom, given that it can be carried in different clothing, different parts of the body, or handled in unknown ways by the user. The challenges in coping with the variations are fundamentally different.

Sensor dead reckoning: Attempts have been made to solve the problem with smartphones, however, solutions make certain assumptions. Specifically, [22, 19] assume that the initial orientation of the phone is known, and the phone is held stable in the hands. Authors in [20] infer orientation by identifying specific gestures like texting, at which point the phone is assumed to be in a known stable orientation. In the outdoor setting, GPS can offer heading direction, although with limited responsiveness; GPS may also falter in Manhattan-like settings. WalkCompass relies only on the direction of the forces on the smartphone, thereby eliminating reliance on other technologies.

Map based: The application of particle filters on floorplans is a popular technique [7, 25, 28, 18, 11]. However, the approach fails in open spaces such as halls, airports, libraries, atriums of hotels, etc. Moreover, the reliance of floorplans restrict the applicability of these solutions. Zee [28] proposes an interesting stand-alone approach to infer the heading direction, essentially deriving hints from the frequency response of the accelerometer. However, the technique alone is insufficient and leaves an ambiguity between two opposite directions. It relies on the map for disambiguation.

Vehicle dynamics: In essence, the analysis of vehicle dynamics using inertial sensors is also related to WalkCompass, despite differences in approach and methodology. Some recent papers in this domain have leveraged sensor data from smartphones to track the movement of vehicles for applications like driver detection [8], driving behavior analysis [15, 31, 10], and estimation of road conditions [23].

Correcting Compass Error

Geomagnetic fields and compass errors are well investigated areas of study – many surveying organizations [6, 26] precisely record and tracks the behavior of global magnetic fields and anomalies. For local magnetic distortions, some mathematical models are available [4] to quantify the error and apply corrective algorithms like *Ellipsoid Fitting* [21]. However, these approaches rely on calibrating the magnetic field at various locations using specific devices and methodology. These techniques are therefore suited for one-time applications, say robotic motion planning or war-driving, but does not scale to anywhere, anytime computing. WalkCompass attempts to correct the compass in a calibration-free, stand-alone manner.

8. CONCLUSION

This paper shows that the human's walking direction can be estimated from the smartphone's inertial sensors, regardless of the orientation of the phone on the body. The core techniques are rooted in analyzing the relationship between human walking and its effect on the phone, as well as methods to estimate and cancel magnetic interference from the compass data. We believe that WalkCompass can immediately help a variety of apps that make assumptions on the user's walking direction. More importantly, we believe that with some more effort, the phone's native compass can be dramatically improved, ultimately helping all applications that rely on the compass.

9. ACKNOWLEDGMENT

We sincerely thank our shepherd, Dr. Xiaofan Jiang as well as the anonymous reviewers for their invaluable feedback. We are also grateful to Intel and NSF for partially funding this research through the following grants: NSF 0910846 and NSF 1040043.

10. REFERENCES

[1] Complementary filter. `http://web.mit.edu/scolton/www/filter.pdf`.

[2] Unibros. `http://hfid.olin.edu/sa2013/s_engr3220-unibros/inspirational.php`.

[3] Walkcompass demo. `http://synrg.csl.illinois.edu/projects/localization/walkcompass`.

[4] Afzal, M. H., Renaudin, V., and Lachapelle, G. Assessment of indoor magnetic field anomalies using multiple magnetometers. *Proceedings of ION GNSS10* (2010), 1–9.

[5] Ayub, S., Heravi, B. M., Bahraminasab, A., and Honary, B. Pedestrian direction of movement

determination using smartphone. In *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on* (2012), IEEE, pp. 64–69.

[6] Barton, C. International geomagnetic reference field: the seventh generation. *Journal of Geomagnetism and Geoelectricity 49* (1997), 123–148.

[7] Chintalapudi, K., Padmanabha Iyer, A., and Padmanabhan, V. N. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 173–184.

[8] Chu, H., Raman, V., Shen, J., Kansal, A., Bahl, V., and Choudhury, R. R. I am a smartphone and i know my user is driving.

[9] Constandache, I., Choudhury, R. R., and Rhee, I. Towards mobile phone localization without war-driving. In *INFOCOM, 2010 Proceedings IEEE* (2010), IEEE, pp. 1–9.

[10] Dai, J., Teng, J., Bai, X., Shen, Z., and Xuan, D. Mobile phone based drunk driving detection. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS* (2010), IEEE, pp. 1–8.

[11] Evennou, F., Marx, F., and Novakov, E. Map-aided indoor mobile positioning system using particle filter. In *Wireless Communications and Networking Conference, 2005 IEEE* (2005), vol. 4, IEEE, pp. 2490–2494.

[12] Faure, F., Debunne, G., Cani-Gascuel, M.-P., and Multon, F. Dynamic analysis of human walking. In *Computer Animation and SimulationâĂŹ97*. Springer, 1997, pp. 53–65.

[13] Gafurov, D., Helkala, K., and Søndrol, T. Gait recognition using acceleration from mems. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on* (2006), IEEE, pp. 6–pp.

[14] Inman, V. T. Human locomotion. *Canadian Medical Association Journal 94*, 20 (1966), 1047.

[15] Johnson, D. A., and Trivedi, M. M. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on* (2011), IEEE, pp. 1609–1615.

[16] Kim, J. W., Jang, H. J., Hwang, D.-H., and Park, C. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems 3*, 1-2 (2004), 273–279.

[17] Krach, B., and Robertson, P. Integration of foot-mounted inertial sensors into a bayesian location estimation framework. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on* (2008), IEEE, pp. 55–61.

[18] Kwon, W., Roh, K.-S., and Sung, H.-K. Particle filter-based heading estimation using magnetic compasses for mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 2705–2712.

[19] Lee, S.-W., Jung, P., and Song, S.-H. Hybrid indoor location tracking for pedestrian using a smartphone. In *Robot Intelligence Technology and Applications 2012*. Springer, 2013, pp. 431–440.

[20] Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., and Zhao, F. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (2012), ACM, pp. 421–430.

[21] Li, Q., and Griffiths, J. G. Least squares ellipsoid specific fitting. In *Geometric Modeling and Processing, 2004. Proceedings* (2004), IEEE, pp. 335–340.

[22] Link, J. A. B., Smith, P., Viol, N., and Wehrle, K. Footpath: Accurate map-based indoor navigation using smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on* (2011), IEEE, pp. 1–8.

[23] Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., and Selavo, L. Real time pothole detection using android smartphones with accelerometers. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on* (2011), IEEE, pp. 1–6.

[24] Merrill, R. T., McElhinny, M. W., and McFadden, P. L. *Magnetic Field of the Earth*, vol. 63. Academic Press, 1996.

[25] Nandakumar, R., Chintalapudi, K. K., and Padmanabhan, V. N. Centaur: locating devices in an office environment. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (2012), ACM, pp. 281–292.

[26] Olsen, L., Major, G., Shein, K., Scialdone, J., Vogel, R., Leicester, S., Weir, H., Ritz, S., Stevens, T., Meaux, M., et al. Nasa/global change master directory (gcmd) earth science keywords. *Version 6*, 0.0 (2007), 0.

[27] Pei, L., Liu, J., Guinness, R., Chen, Y., Kuusniemi, H., and Chen, R. Using ls-svm based motion recognition for smartphone indoor wireless positioning. *Sensors 12*, 5 (2012), 6155–6175.

[28] Rai, A., Chintalapudi, K. K., Padmanabhan, V. N., and Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (2012), ACM, pp. 293–304.

[29] Susi, M., Renaudin, V., and Lachapelle, G. Motion mode recognition and step detection algorithms for mobile phone users. *Sensors 13*, 2 (2013), 1539–1562.

[30] Wang, Y., Jia, X., Lee, H., and Li, G. An indoors wireless positioning system based on wireless local area network infrastructure. In *6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services* (2003), no. 54.

[31] Wang, Y., Yang, J., Liu, H., Chen, Y., Gruteser, M., and Martin, R. P. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services* (2013), ACM, pp. 41–54.

[32] Woodman, O., and Harle, R. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing* (2008), ACM, pp. 114–123.