

Walking compass with head-mounted IMU sensor

Jens Windau and Laurent Itti, University of Southern California

Emerging wearable technologies offer new sensor placement options on the human body. Particularly, head-mounted glass-wear opens up new data capturing possibilities directly from the human head. This allows exploring new cyber-robotics algorithms (robotics sensors and human motor plant). Glass-wear systems, however, require additional compensation for head motions that will affect the captured sensor data. Particularly, pedestrian dead-reckoning (PDR), activity recognition, and other applications are limited or restricted when head-mounted sensors are used, because of possible confusion between head and body movements. Thus, previous PDR approaches typically required to keep the head pointing direction aligned with the walking direction to avoid positional errors. This paper presents a head-mounted orientation system (HOS) that identifies and filters out interfering head motions in 3 steps. Step 1 transforms inertial sensor data into a stable normalized coordinate system (roll/pitch motion compensated). Step 2 compares walking patterns before and after a rotating motion. Step 3 eliminates interfering head motions from sensor data by dynamically adjusting the noise parameters of the extended Kalman filter. HOS has been implemented on a Google Glass platform and achieved high accuracy in tracking a person's path even in the presence of head movements (within 2.5% of traveled distance) when tested in multiple real-world scenarios. By eliminating head motions, HOS not only enables accurate PDR, but also facilitates the task for downstream activity recognition algorithms.

I. INTRODUCTION

The industry for smart personal devices with integrated sensors suitable for position tracking and activity recognition has been emerging over the last few years. While this trend began first with Smartphones, various devices such as smart watches, fitness trackers, and increasingly head-mounted devices are now entering the market. Prominent examples are Google Glass, Microsoft's HoloLens and Oculus Rift. These head-mounted devices are equipped with inertial sensors (IMU). While wearing them, the user is constantly recording valuable sensor data. This information can be used to make this technology smarter by performing activity recognition and creating situation awareness for the device, which can then adapt its behavior to the current user activity. This could greatly enhance many device applications, e.g., music streaming with Google Play Music: for different activities (e.g., working out, making food, cleaning the house, or driving a car), the service adapts its music offerings; but currently users have to manually indicate which activity they are engaged in. We showed in previous work that head-mounted sensor systems can successfully distinguish between many daily activities through online analysis and classification of the streaming IMU data, but this was when the head was held pointing straight forward [10]. Head-mounted devices can also be used to track motions with-

out external reference systems in 3D Augmented Reality Virtual Rooms. This allows users to create training simulations, exercise in environments or help with home improvement by visualizing remodeling projects (e.g., Lowe's Holeroom), with again the current problem that head and body motions may be confused.

Tracking motions and activities is difficult. The head pointing direction may not always align with the body orientation, which can lead to a wrongly classified activity or a positional error in PDR. For this reason, we developed a stabilization algorithm that enables all these applications. To test and validate our work, we use PDR as one of the most stringent applications, and we focus on being able to recover walking direction and trajectory over time even as the head occasionally turns side to side [figure 1].

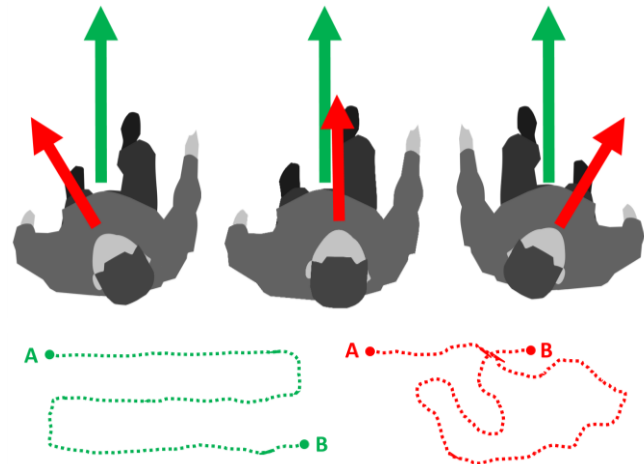


Figure 1: Possible head pointing directions (red arrows) and walking directions (green arrows). Head-mounted pedestrian dead-reckoning systems need to compensate for head motions for correct position tracking (green path) and to avoid false positional tracking (red path).

PDR has been well studied for various sensor systems and is mainly used to supplement other navigation methods. PDR systems have been applied to many use-cases with different placements of sensors on most body parts (helmet, foot, backpacks, pockets, wrist, and legs). Most PDR systems work after the same principle: Identifying footsteps via accelerometer and aligning them to the proper heading direction via orientation sensors such as gyroscope and compass. A team of the University of Bremen developed a safety helmet for emergency services. IMU sensors and a GPS receiver attached to the helmet were used for indoor/outdoor navigation. Errors were as large as 5.4% of the total distance travelled with a cumulated model step and 2% with a neural network prediction [1] [2]. However, the helmet needs to be pointed in the direction of walking at all times and cannot compensate for head motions [1]. IMU sensors have also been attached to shoes in several projects.

Jens Windau and Laurent Itti are with the iLab and Computer Science Department at the University of Southern California, Los Angeles, CA, 90089-2520, USA. (Contact Information: website: <http://ilab.usc.edu> phone: +1(213)740-3527 email: jenswindau@gmail.com, itti@pollux.usc.edu)

The MiPN Project focused on supporting a soldier's GPS system, particularly for indoor tracking. The NavShoe System was developed for firefighters or other emergency first responders. In both cases, the sensor data was merged with GPS data for long-term drift compensation. In general, the position error for foot-mounted sensors varies between 0.3% - 2% of the distance traveled [4] [7], but this requires an additional battery-powered device on the foot. Smartphone-based indoor navigation systems reached 0.19% - 2.92% accuracy depending on the smartphone's placement [5] [6] [8]. All tested PDR systems only work reliably if the orientation of the sensors aligns with the orientation of the walking direction. PDR systems have also assisted vision systems and decreased positional error to 0.5%. Navigation applications can be found in the tourism industry, particularly guiding tourists through museums, train and subway stations, and large buildings. Blind or visually impaired people could also benefit from a navigation assistant [9]. For instance, the Walking Compass tries to identify walking direction from accelerometer patterns [3]. Results show different performances depending on the position the smartphone is carried in: Their accuracy is $\pm 5^\circ$ (palm), $\pm 3^\circ$ (pant pocket), $\pm 8^\circ$ for swinging-hand. However, errors were reported when user vary the phone's orientation during operations. Instead of aligning footsteps with the heading, one could also double integrate the acceleration. However, due to the complex composition of acceleration data (gravitational acc., linear acc., coriolis acc., centrifugal acc., static noise, and acceleration bias), double integration is not a suitable technique for low-cost IMU sensors since it requires very accurate linear acceleration data. Overall, the performance of current PDR systems is promising, however, many restrictions apply particularly for the orientation of the sensors. Short-term errors are related to incorrect heading directions [2] that can be caused by rotational motions (body, head, arm). Thus most applications that do not have efficient compensation methods will be restricted to keep their sensor orientation in a fixed orientation (e.g., perpendicular) to the walking direction [5]. This is why no head-mounted PDR system has been presented yet, which actively compensates for rotational head-motions and performs comparably to existing PDR systems. The HOS approach presented here is based on head-mounted IMU data and has a head-motion compensation algorithm. This algorithm can effectively compensate head-motions as long as they do not overlap with rotational body motions. Synchronous rotational movements of head and body lead to a confusion and require additional information or complementary sensor data (e.g., GPS tracking over time) to recover the correct orientation information. The HOS approach is described in detail in the following sections, and it has a set of unique contributions: (1) analysis of head-mounted IMU data during walking and head movements, (2) effective segmentation technique of rotational motion sensor data, (3) selection of critical features for filter tuning, (4) fusion algorithm with dynamical tuning, (5) collection of test scenarios.

II. ANALYSIS OF IMU DATA DURING WALKING ACTIVITY

To understand the frequency composition of walking, 60 seconds of X-Y plane accelerometer sensor data was recorded. During the recording, the head pointed in multiple directions within a $\pm 60^\circ$ range from the walking direction (figure 2). The sensor data was then divided into multiple 3 sec windows and transformed into the frequency spectrum. Two frequencies

showed a critical magnitude (figure 3). The frequency value of the peaks depends on the walking speed (and frequency of footsteps), but can be classified as:

Side Swinging Peak: Body swings to the right and left side for every step taken. The accelerometer will capture the swing motion direction change. One full swing sequence requires two steps.

Stepping Peak: Each step generates linear acceleration front-to-back (in addition to a vertical component). When walking with the head straight (aligned with walking direction), steps are dominantly recorded by the x axis, while swing is captured by the y axis. When walking with the head pointed sideways, both x and y axes measure both stepping and swinging. With the head turned 90° , the peaks would be measured by the reverse axes (swinging by x, stepping by y).

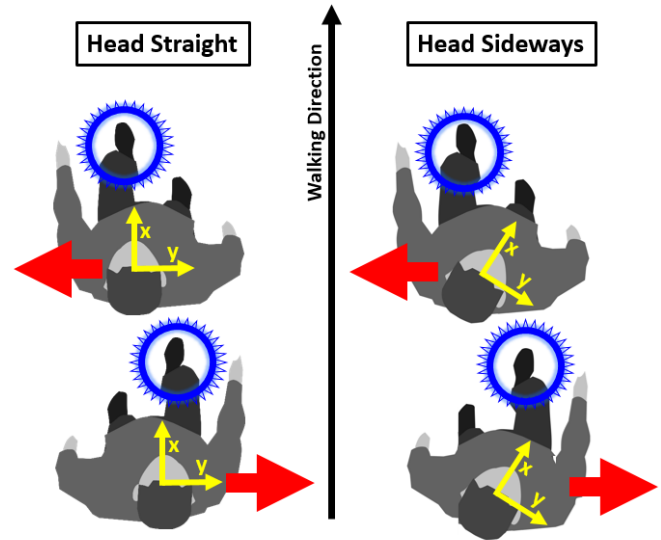


Figure 2: X/Y axis detect swinging and stepping in different magnitude depending on head orientation.

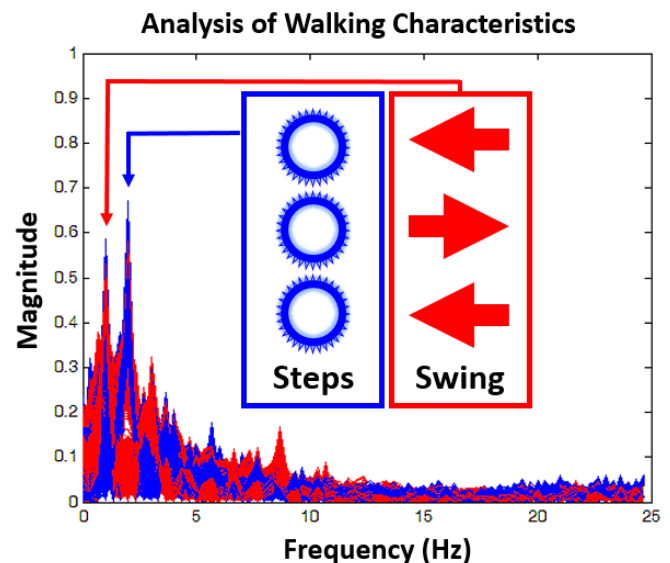


Figure 3: Frequency Spectrum of walking activity with head turned in multiple orientations.

III. THE MEASUREMENT DETAIL

A. Overview

HOS records IMU data at 50 Hz. The Google Glass platform provides the accelerometer data as well as an orientation estimator for Roll and Pitch. The architecture is structured around the 3 major processing steps summarized in figure 4.

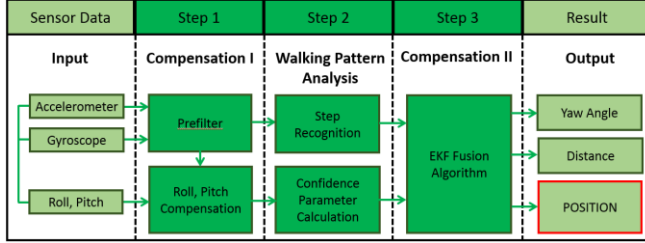


Figure 4: Overview of the system architecture

Step 1 pre-filters sensor data and transforms it into a stable normalized coordinate system to compensate for roll and pitch changes. This is necessary to measure the accelerometer data in a stabilized X/Y plane. Step 2 identifies walking patterns among the accelerometer axis x and y. A step recognition algorithm determines the step duration from the walking patterns by detecting peaks in the acceleration that correspond to foot-falls. Head motions can be identified by changes in the walking pattern described by a confidence parameter. Step 3 uses this parameter to eliminate gyroscope data from head-motions. The result of the 3 steps is the yaw angle which describes the heading direction of the pedestrian. If combined with step counts of Step 2, the position of a person can be tracked.

B. Steps in Detail

Step 1: Compensation for Roll and Pitch Motions

In this step, the raw IMU sensor data of the Google Glass platform is first pre-filtered (offset compensated to minimize gyro drift). Then, the data is transformed into the normalized coordinate system. This transformation is necessary to measure two critical types of sensor data during walking independently of the actual head orientation [10].

(1) The angular velocity ω_z around the z-axis (vertical axis, perpendicular to the ground plane) which is used to calculate the heading direction ϕ .

(2) The acceleration a_x and a_y along the x and y axis (in the plane parallel to the ground plane) which is used to calculate the confidence parameter C_{gyro} for the gyroscope data.

The transformation is accomplished by two equations (1) (3), where a is a 3x1 vector representing the accelerometer sensor data and ω is a 3x1 vector representing the angular velocities.

$$a_{normalized} = R_a^{-1} \cdot a_{sensor} \quad (1)$$

$$R_a = R_x(\phi)R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\theta)\sin(\phi) & \cos(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \quad (2)$$

$$\omega_{normalized} = R_\omega^{-1} \cdot \omega_{sensor} \quad (3)$$

$$R_\omega = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (4)$$

Note that this formulation using Euler Angles suffers from a singularity at $\theta = \pm 90^\circ$ which is not an issue here because it relates to an unrealistic head pose. The IMU data is now compensated for roll and pitch motions. Only the normalized z-axis of the IMU is aligned with the gravity vector. The normalized x- and y- axis measure sensor data without any gravity component. In this gravity-compensated coordinate system, the sensor data is now used for the walking pattern analysis.

Step 2: Walking Pattern Analysis

The HOS-Filter measures changes in the walking pattern as captured in the accelerometer data. These changes are used to calculate a confidence parameter on how much the HOS-Filter can trust the gyroscope data. This confidence parameter is the only critical feature to evaluate a detected rotational motion.

To accomplish step 2, the transformed gyroscope is used to identify moments of rotational motions. These timestamps are split at segmentation points, then analyzed in a Gyroscope Activity Zone and an Observation Zone.

Segmentation Points: The segmentation points (red big arrow in figure 5 and 6) are segmented by a necessary minimum peak height (absolute value) of normalized gyroscope data as well as necessary minimum peak distance.

Gyroscope Activity Zone: The gyroscope activity zone is an isolation zone around the segmentation point that involves gyroscope activity and must not be used for evaluation of the walking pattern. The algorithm extracts a time-window around a segmentation point until the normalized gyroscope data falls below a set threshold. The gyroscope activity zone only tells us if the gyroscope captured a rotational motion. It does not tell us if the head direction or the walking direction changed.

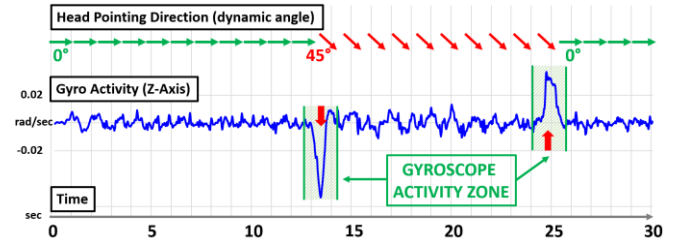


Figure 5: Walking a straight line with changing head pointing direction. Gyroscope activities with a peak threshold will be captured (red arrow) and the gyroscope activity zone segmented (green area)

Observation Zone: The observation zone is necessary to evaluate the walking pattern before and after the gyroscope activity zone. It will create the most critical feature to calculate the “confidence” level of gyroscope data. Two fixed-length observation zones are extracted, one before (B) and one after (A) a

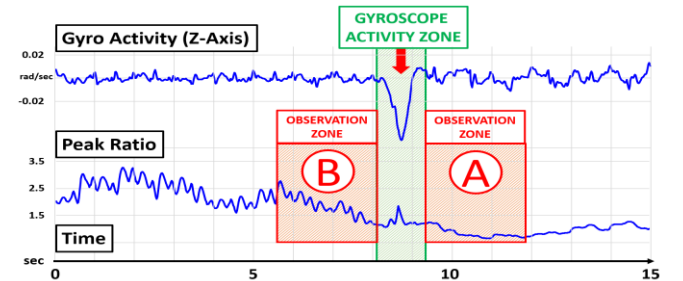


Figure 6: The Peak Ratio will be observed before (B) and after (A) the Gyroscope Activity Zone and will provide valuable information if the walking pattern changed

gyroscope activity zone (figure 6). These observation zones are up to 180 samples large (3.6 sec), they are compared to each other and analyzed for similarity. The similarity is shown as a confidence parameter (1 = highest similarity, 0 = no similarity). The larger the similarity, the less likely a head motion occurred, since the alignment of the head to the walking pattern remains unchanged between A and B. To calculate the similarity, we need to first measure the Peak Ratio P_{ratio} .

$$\text{Peak Ratio } P_{ratio} = \frac{P_{X1}}{P_{X2}} \quad (5)$$

This value is calculated by dividing the magnitude of P_{X1} by P_{X2} . Both parameters P_{X1} and P_{X2} represent the two critical magnitude peaks of the FFT-transformed accelerometer data window of the x-axis in the normalized coordinate system. Dividing P_{X1} by P_{X2} works amplifying on the final Peak Ratio. Once one magnitude increases (P_{X1} or P_{X2}), the other one declines (P_{X2} or P_{X1}). The Peak Ratio represents the ratio between swinging and stepping activity and is directly related to the head pointing direction relative to the walking direction. Depending on the head angle, the peak ratio changes accordingly (figure 7).

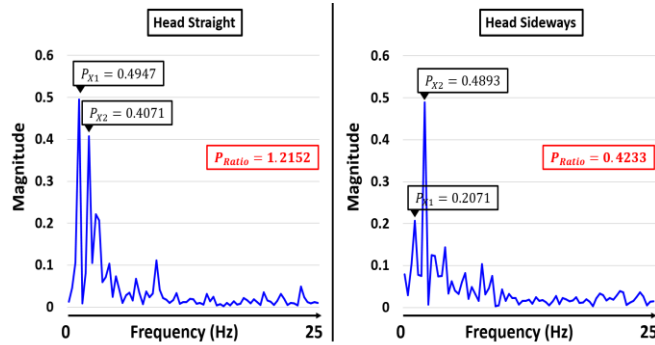


Figure 7: Calculation Example for Peak Ratio with Head Straight and Head Sideways

Next, peak ratio data of the observation zones are selected. This data is necessary to evaluate the walking pattern before and after the gyroscope activity zone. For both zones, the expected value μ and the standard deviation σ are calculated and used to calculate the confidence parameter. This parameter describes the trust in the gyroscope data. The confidence parameter equation is derived from [11] and approximates the intersection area of the two Gaussian Bell curves. The confidence parameter is calculated with the Complementary Error Function erfc . The max confidence is 1 for a max intersection area. The min confidence is 0 for no intersection area.

$$\text{Confidence Parameter } C_{gyro} = \text{erfc}\left(\frac{|\mu_1 - \mu_2|}{\sqrt{2 \cdot (\sigma_1^2 + \sigma_2^2)}}\right) \quad (6)$$



Figure 8: The confidence parameter approximates the intersection area of the two Gaussian Bell Curves. Both curves represent the Observation Zone before and after the gyroscope activity.

Step 3: Fusion Algorithm

The system uses a fusion algorithm to combine IMU measurement data (accelerometer, gyroscope) with the derived features: step size (assuming const. 71.6 cm length), walking velocity (step size / step duration), and the confidence parameter (figure 9). The confidence parameter is the only feature to evaluate if the gyroscope data is counted as a head motion or not. If the gyroscope detects a rotational motion, the system dynamically adjusts the measurement noise matrix. For this step, the confidence parameter plays an important role since it describes the ‘trust’ in the gyroscope data. A large parameter expresses high trust (no head motions). Barely or no intersection show low ‘trust’ into the gyroscope data. This information will then be used for the Extended Kalman Filter Fusion Algorithm (figure 9).

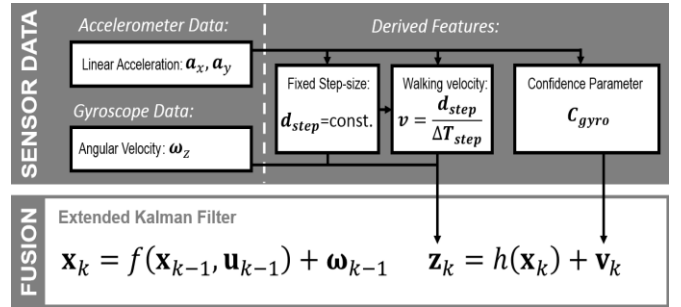


Figure 9: Fusion Algorithm to combine IMU Data with derived features.

A Constant Velocity / Constant Angular Velocity Model (CV ω -Model) was developed. The model is described with two state dynamics equations. The first is called the state transition model. This equation calculates the system state vector x_k by the system’s dynamic function $f(x_{k-1}, u_{k-1})$ with u_{k-1} as the control vector and added process noise ω_{k-1} . The second equation describes the observation model. The observation vector z_k is calculated by the measurement function $h(x_k)$ and the measurement error source v_k . The Extended Kalman Filter uses two steps to fuse the sensor data: The prediction step and the update step.

(1) Prediction Step

The state estimate x of the pedestrian includes the positional information x and y , the heading angle φ , and the absolute velocity v . The CV ω -Model does not require the heading angle φ to be constant. This model only assumes that the angular velocity ω (turning rate of the pedestrian), will stay constant. In addition the absolute velocity is measured via the step detection and as well remains constant in the CV ω -Model. The following equation describes the implementation of the system’s dynamic function $f(x_{k-1}, u_{k-1})$.

$$x_k = \begin{bmatrix} x \\ y \\ \varphi \\ v \\ \omega \end{bmatrix}_{k|k-1} = \begin{bmatrix} x \\ y \\ \varphi \\ v \\ \omega \end{bmatrix}_{k-1|k-1} - \begin{bmatrix} vT \cos(\varphi) \\ vT \sin(\varphi) \\ \omega T \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

(2) Update Step

The first part of the update step is calculating the innovation \tilde{y}_k by subtracting the predicted measurement $h(x)$ from the sensor measurement.

$$\tilde{y}_k = \begin{bmatrix} r \\ r \\ \omega_{gyro} \end{bmatrix}_k - \begin{bmatrix} \sqrt{(vT\cos(\varphi))^2 + (vT\sin(\varphi))^2} \\ v \\ \omega \end{bmatrix}_{k|k-1} \quad (8)$$

The measurement noise matrix R of the fusion filter is dynamic. It is adjusted in every update cycle with the confidence parameter C_{gyro} of the gyroscope data. Since the noise level should decrease with increasing confidence parameter and vice versa, C_{gyro} needs to be subtracted from 1. It is multiplied with a tuning factor k to adjust its range for the fusion filter.

$$\text{Dynamic Noise Matrix: } R = R_{static} + R_{dynamic} \quad (9)$$

$$\text{Dynamic component: } R_{dynamic} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k \cdot (1 - C_{gyro}) \end{bmatrix} \quad (10)$$

$$\text{Static component: } R_{static} = \begin{bmatrix} \sigma_{mr}^2 & 0 & 0 \\ 0 & \sigma_{mv}^2 & 0 \\ 0 & 0 & \sigma_{m\omega}^2 \end{bmatrix} \quad (11)$$

If the belief C_{gyro} in the gyroscope data is high (EKF noise is low), the fusion filter will trust the gyroscope data and validate gyroscope activity as changes in the walking direction. If the trust in C_{gyro} is low (EKF noise is high), the fusion filter will consider the gyroscope data related to head-motions and will not take it into account. The output of the Extended Kalman Filter is a vector of five entries including the position and heading angle of the pedestrian.

IV. TEST RESULTS

A. Performance Tests of HOS

Two types of tests were performed on a university campus parking lot. The first test includes a basic case test evaluating the performance of three base scenarios. The second test includes a more diverse testing setup with multiple base scenarios combined. Each test scenario was repeated three times. We analyzed a total of 12 datasets. We are particularly interested in the short-term detection since of the nature of PDR systems assisting GPS systems and other navigation systems that provide a long-term stability.

B. Basic Case Testing

HOS was tested with datasets of 30 sec of one subject. Each dataset counts around 50 steps describing a specific case. In the case testing, the success rate of the Filter Algorithm was tested for every case independently.

Setup: The performance of HOS was first tested on 3 basic cases (figure 10): **Case 1** (Walking straight line, no head direction change), **Case 2** (Walking straight line, with head direction change), and **Case 3** (Walking with turn, no head direction change).

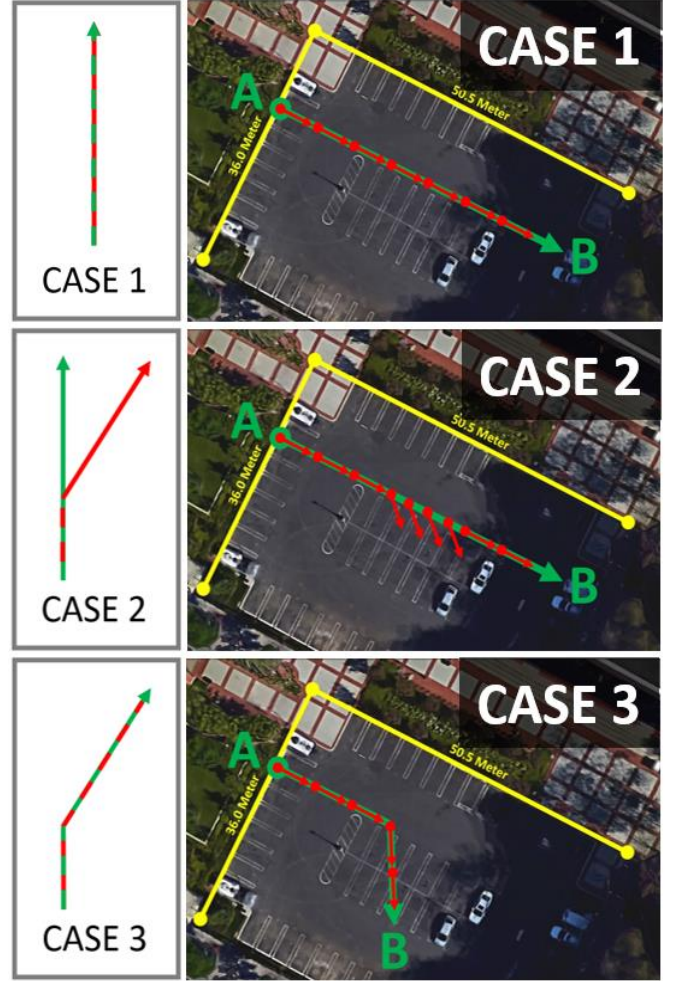


Figure 10: Three basic case scenarios (Red Arrow: Head pointing direction, Green Arrow: Walking Direction)

Results

Case 1 and Case 3 – no head motions involved

For Case 1 and Case 3, the results are expected to be similar to current PDR systems, since the head pointing direction aligned with the walking directions at all times. Like previous PDR systems we measure error as a percentage of distance traveled (figure 11). When a walking direction change occurred, the filter algorithm successfully validated the rotational activity as a change in walking direction and took all rotational motions into account.

Case 2 – head motions involved

For Case 2, the filter algorithm successfully detected head-motions and disregarded the gyroscope activity. Standard PDR systems would fail in this case scenario since the change in the head pointing direction would have been validated as a change in the walking direction. The error for standard PDR systems would increase depending on the heading angle error $\Delta\varphi$:

$$\text{PDR Error} = \sqrt{2 \cdot (1 - \cos(\Delta\varphi))} \quad (12)$$

Case	Walking Direction	Head Pointing Direction	PDR Error*	HOS Error
1	No changes	No changes	<2.5%	<2.5%
2	No changes	Rotation 47° 48° 35°	73.7%	<2.5%
3	Rotation 25° 35° 45°	No changes	<2.5%	<2.5%

Figure 11: Results of the Basic Case Testing. (*) PDR error from system in [1] [2] [5] [6] [8].

C. Mixed Case Testing

Setup

HOS was tested with three datasets containing data of a walking subject of 30 sec. Each dataset counts 50 steps (35 meters) and is a combination of multiple base cases (figure 12). In the scenarios, the success rate of the Filter Algorithm was tested.



Figure 12: Motion sequence for mixed case testing

Results



Figure 13: Combination of Case 1, Case 2, and Case 3. The blue path represents the walking path, the white arrows are the head pointing directions. The red markers show head motion activity, the green marker represents a walking direction change

HOS successfully identified and filtered the head-motion from the recorded data (figure 13). HOS performed an error lower than 2.5% of the traveled distance. Standard PDR systems would have already gained an error of approximately 9.1 meters due to the left head turn which would have been misclassified as a left body turn. Since this error would further grow over time, navigation would not be possible (figure 14).

Case	Walking Direction	Head Pointing Direction	PDR Error*	HOS Error
1+2+3	90° Rotation	Rotation 46° 52° 55°	26%	<2.5%

Figure 14: Results of Mixed Case Testing (*) PDR error from system in [1] [2] [5] [6] [8].

V. FUTURE IMPROVEMENTS AND CONCLUSION

We have shown that interfering head motions can be successfully detected and compensated for in PDR applications. However, in real-world scenarios, head rotations and body rotations may occur at the same time or head motions could be unrecognized when performed gradually over a longer time. For this work, the walking pattern provided critical information about whether a head-motion occurred. To do motion tracking performed by both head and body at the same time, more features besides the confidence parameter (e.g. rotational acceleration may exhibit different patterns during head turns and body turns) need to be extracted from the sensor data and walking pattern. These features would need to be tested on many subjects who are both walking and moving their head in unconstrained real-world situations. In addition, head tracking can be performed during many activities. In future research, we will focus on showing that the HOS algorithm is also suitable for different activities with recurring patterns such as jogging, running, biking, and skating. By providing a valuable approach for PDR from a head-mounted sensor system, we hope that HOS made a contribution to the development of navigation methods via wearable technology.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (grant number CCF-1317433), and the Office of Naval Research (N00014-13-1-0563). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

REFERENCES

- [1] S. Beauregard, "A Helmet-Mounted Pedestrian Dead Reckoning System", 3rd International Forum on Applied Wearable Computing (IFAWC), Bremen, Germany, 2006, pp. 1 – 11
- [2] Y. Jin, H.-S. Toh, W.-S. Soh, W.-C. Wong, "A Robust Dead-Reckoning Pedestrian Tracking System with Low Cost Sensors", IEEE International Conference on Pervasive Computing and Communications (perCom), Seattle, 2011
- [3] N. Roy, H. Wang, R.R. Choudhury, "I am a Smartphone and I can Tell my User's Walking Direction", Proceedings of the 12th annual international conference on Mobile systems, applications, and services (MobiSys), 2014, PP. 329-342
- [4] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors", IEEE Computer Graphics and Applications, 2005, pp. 38 – 46
- [5] W. Kang, S. Nam, Y. Han, S. Lee, "Improved Heading Estimation for Smartphone-Based Indoor Positioning Systems", IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications – (PIMRC), 2012
- [6] D. Pai, M. Malpani, I. Sasi, P. S. Mantripragada, N. Aggarwal, "Padiati: A Robust Pedestrian Dead Reckoning system on smartphones", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communication, 2012
- [7] J. Bird, D. Arden, "Indoor Navigation with Foot-Mounted Strapdown Inertial Navigation and Magnetic Sensors", IEEE Wireless Communication, 2011
- [8] A. R. Pratama, Widyawan, R. Hidayat, "Smartphone-based Pedestrian Dead Reckoning as an Indoor Positioning System", International Conference on System Engineering and Technology, 2012
- [9] M. Li, B. H. Kim, A. I. Mourikis, "Real-time Motion Tracking on a Cellphone using Inertial Sensing and a Rolling-Shutter Camera", ICRA, 2013
- [10] J. Windau, L. Itti, "Situation awareness via sensor-equipped eye-glasses", IROS, 2013
- [11] L. Itti, "Models of Bottom-Up and Top-Down Visual Attention", PhD Thesis, Caltech, Pasadena, CA, USA, 2000, pp. 149