Usability of Smartphone Inertial Sensors for Confined Area Motion Tracking

Anton Umek, Anton Kos
University of Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia
E-mail: anton.umek@fe.uni-lj.si,
anton.kos@fe.uni-lj.si

Abstract - *Modern smart phone devices are equipped with several space positioning sensors. Most of them are inaccurate low-cost silicon devices, not designed for motion tracking. The paper presents the results of several constraint motion tracking experiments using iPhone 4 sensors. The experiments confirm that the best choice for motion tracking is sensor fusion - a simultaneous usage of accelerometer and gyroscope data. While accelerometer data are less accurate than gyroscope data, they are still good enough for a number of various motion-connected applications.*

1 Introduction

Inertial sensors in iPhone 4 are embedded in two IC devices manufactured by STMicroelectronics: 3D accelerometer LIS331DLH [2] and 3D gyroscope L3G4200D [3]. Devices are not labeled with original part numbers, but identified by Chipworks [1]. Both devices are designed more for movement detection, gaming and virtual reality input devices and less for navigation applications. The major sensor parameters are listed in table 1.

Parameter	LIS331DLH Accelerometer	L3G4200D Gyroscope
Measurement range	$\pm$ 2 g	$\pm$ 2000 deg/s
Sensitivity	1 ± 0.1 mg/dig	70 mdeg/s/dig
Bias error	$\pm$ 20 mg	$\pm$ 75 deg/s

Table 1. iPhone 4 inertial sensor parameter values.

Both biases induce deviations on the derived spatial and angular position. The relative space position is calculated by integrating the acceleration vector twice over time. For a simplified one dimensional motion, the position error is equal to the path length error Δs. The acceleration bias Δa makes a linear drift in velocity and a squared drift in position:

$$\Delta s = \frac{1}{2} \Delta a \, t^2 \tag{1}$$

The rotation angle is calculated by integrating the angular velocity over time. The gyroscope bias makes a linear drift: $\Delta\alpha = \Delta\omega \, t$. The position accuracy is more sensitive to acceleration bias, but in longer time periods both drifts blur the actual position. Therefore both sensor biases should be compensated.

2 Motion constraints and tracking

Our experiments include two important constraints, which simplify motion trajectory calculation:

- the motion of the device is in a two dimensional well-balanced horizontal plain, perpendicular to the gravitational vector,
- the motion orientation of the device is always in sensor's principal direction 1_y.

Assuming there is no gravitation projections in plain dimensions (x,y), only the accelerometer bias requires compensation in directions: Δa_x, Δa_y. Assuming there is no sideway slithering, smart phone velocity vector absolute value depends only on the acceleration vector component a_y. If all of the above conditions are fulfilled, two simple tracking algorithms can be used.

Algorithm 1

The algorithm uses accelerometer data a_y and gyroscope data ω_z. The acceleration bias Δa_y should be compensated, otherwise it has the same influence on the path length error Δs as in a one dimensional motion (1). The gyroscope bias $\Delta\omega_z$ makes a linear drift in the device orientation $\alpha[n]$ and should be compensated.

The starting velocity $v[0]$ and orientation $\alpha[0]$ of the device should be defined. The velocity vector is calculated by sensor fusion: the velocity $v_d[n]$ is calculated from the acceleration vector component a_y and the orientation $\alpha[n]$ is obtained from gyroscope data ω_z:

$$v_d[n] = v_d[n-1] + T_s(a_y[n] - \Delta a_y) \tag{2}$$
$$\alpha[n] = \alpha[n-1] + T_s(\omega_z[n] - \Delta\omega_z) \tag{3}$$

where T_s is sensor data sampling time.

Algorithm 2

The algorithm uses only accelerometer data (a_x, a_y). Both accelerometer bias values should be compensated. The starting velocity vector should be defined $v_{xy}[0] = (v_x[0], v_y[0])$. The orientation of the device $\alpha[n]$ is obtained from the current velocity direction angle. The velocity difference vector $dv_{xy}[n]$ is calculated from the device acceleration vector, obtained from the accelerometer data $a_{xy} = (a_x, a_y)$:

$$dv_{xy}[n] = T_s(a_{xy}[n] - \Delta a_{xy})^{Rot(\alpha[n]-\frac{\pi}{2})} \quad (4)$$

$$v_{xy}[n] = v_{xy}[n-1] + dv_{xy}[n] \quad (5)$$

where $Rot(angle)$ assigns 2D rotation of the device to the absolute coordinate system. The starting device orientation angle $\alpha[0]$ is measured relative to the principal sensor direction 1_y. The relative device position and the path length are measured from the starting position $r[0] = (x[0], y[0])$, $d[0] = 0$:

$$r[n] = r[n-1] + T_s v_{xy}[n] \quad (6)$$

$$d[n] = d[n-1] + T_s|v_{xy}[n]| \quad (7)$$

Instead of using two dimensional matrix algebra, complex numbers can simplify numeric calculations.

Both 2D tracking algorithms are used in our experiments. The first algorithm gives much better motion tracing results than the second algorithm, where gyroscope data is not used. The main reason for such results is that the orientation obtained from two noisy and biased accelerometer components is very inaccurate, especially when accelerometer readouts are low. Gyroscope data has been recognized as very accurate; in comparison to accelerometer bias, relatively small gyroscope bias value $\Delta\omega_z$ does not have a significant effect on the device motion tracking results.

3 Experiments

All experiments were done under the constraints specified in the previous section: the smart phone is horizontally balanced in the (x, y) plane, which is perpendicular to the earth's gravity vector, and the motion of the smart phone is always in sensor's principal axis 1_y. Among several experiments we have chosen the two most representative examples. Their settings are shown in figures 1 and 2. We did not use any dedicated mechanical laboratory equipment.

Several experiments were done using a simple kid's toy; a LEGO City train set, which is flexible and precise enough to build different tracks. Two testing track configurations are illustrated in figure 1. The results that follow in the next section correspond to the inner track. The train composition itself is not shown, but it is easy to precisely install a smart phone onto it in a way that ensures its motion sensors are in the middle of the track. The train is driven by a simple start/stop remote control.

Some experiments were done by mounting the smart phone onto a bicycle front wheel, as illustrated in figure 2. The wheel was accelerated by hand from resting position for approximately 90 degrees and after ten free-drive rotations smoothly braked and stopped at the starting position angle.

Sensor data is recorded by iPhone application *Sensor Monitor Pro* and later processed on a PC.

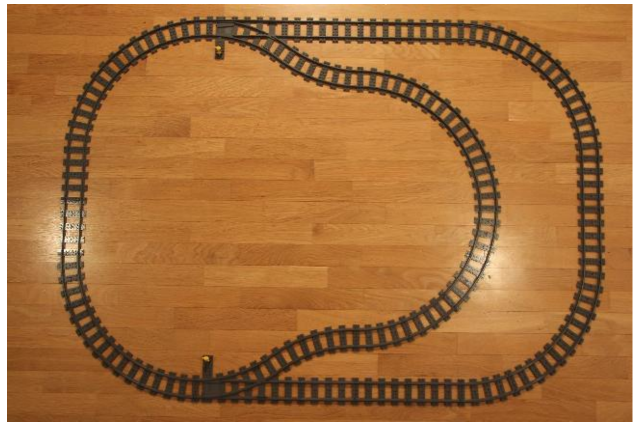

Figure 1. A LEGO City train track. The outer track is in the shape of a rounded rectangle and the inner track is in the shape of a "babuška". More interesting is the inner track that changes the course of the train several times in both directions (left/right). Hence its accelerometer and gyroscope sensor readings are more diverse and interesting for analysis (see figure 3 for details). The smart phone is mounted onto the train in the position that ensures its sensors are in the middle of the track.

Figure 2. A horizontally balanced bicycle wheel. The smart phone is tied to the wheel. When the wheel is spinned, the sensors go round the wheel's axes in a circle with the radius of $r = 22$ cm.

4 Results

Recorded sensor data was transferred to the PC and processed by both algorithms from section 2 using different pairs of sensor signals (a_y, ω_z) and (a_x, a_y). Figure 3 shows signals of the first experiment.

If accelerometer and gyroscope are used for motion tracking, then both offset values for a_y and ω_z should be tuned well in order to fit the motion trajectory close to the known trace pattern of the experiment. Accelerometer and gyroscope biases have independent influence on different velocity parameters (2), (4). Off-

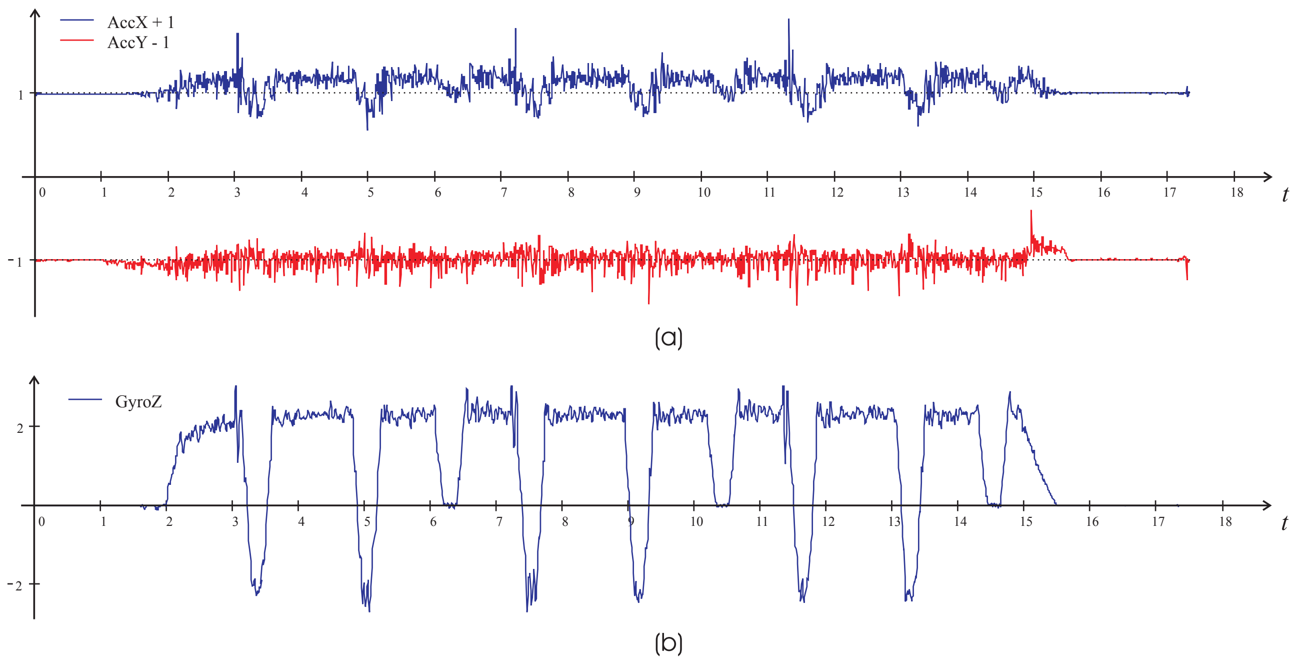

Figure 3. Smart phone sensor readings for the inner track from figure 1. (a) accelerometer readings in X axis (accX) and Y axis (accY), both normalized to the earth's gravity $g = 9.81$ m/s^2. For the purpose of the presentation accX readings are raised and accY readings are lowered for 1 g to prevent the overlapping. (b) gyroscope readings around Z axis (gyroZ). Since all experiments were conducted with the phone balanced in the (x, y) plane, all other readings from the accelerometer and gyroscope are insignificant for our results. AccZ shows earth's gravity, gyroX and gyroY are zero as the phone can only rotate around Z axis.

set values can be fitted separately. Accelerometer and gyroscope offsets should be adjusted to prevent velocity (magnitude) and angular drift. Many details of the motion of the device in experiments are known and can be used in sensor data post-processing.

Gyroscope and accelerometer bias values are first measured by averaging signals in the resting time interval before acceleration. Measured gyroscope bias, averaged over 100 samples, is $5\,10^{-3}$ rad/s. The predicted angular drift in a 15 s long experiment is less than 5 degrees and below the actual trajectory measurement tolerance.

Unfortunately, the averaged measured accelerometer bias is not accurate enough to compensate the velocity drift. The adjusted zero-drift offset is $6.2\,10^{-3}\,g_0$ and differs from the averaged measured bias for more than 30%. The effect of accelerometer bias compensation is illustrated in figure 4. If only accelerometer signals are used for the 2D motion tracking then both biases influence the velocity angular error. Both trajectory best-fit offset values were set very close (± 5%) to averaged measured biases.

Comparison of both methods, using different inertial sensor signals from the first experiment, is illustrated and explained for velocity, orientation and path profile in figures 5 ,6 and 7. Some minor differences are visible between the cumulative path lengths and velocity magnitudes, while angular differences are much higher. Both motion trajectories are compared in figure 8.

Better accuracy in motion tracking was found in the second experiment shown in figure 2. In figure 9 we present only the calculated trajectory with more accurate algorithm 1, where both sensor data are used. The trace path makes almost perfect circles with little offset in the central point.

Accelerometer offsets vary with experiments, sometimes even when measured in resting time before or after the acceleration. The measured accelerometer offset values are inside the specified tolerance from table 1. The measured angle drifts are very low in comparison with gyroscope sensor specification in table 1. Sensor monitoring application obviously uses iOS filtered gyroscope data, where sensor fusion algorithm already compensates large amount of gyroscope bias. The later also explains why measured gyroscope offsets vary in different experiments. Similar results are reported in related work by several authors [4], [5], [6].

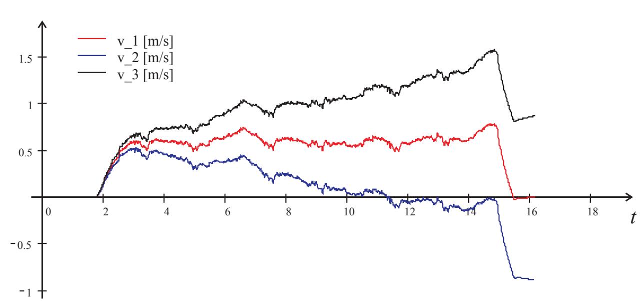

Figure 4. Smart phone velocities in the direction of Y axis v_y in three different cases. Curves show v_y when a_y bias is: compensated (v_1), not compensated (v_2), and with double offset (v_3).

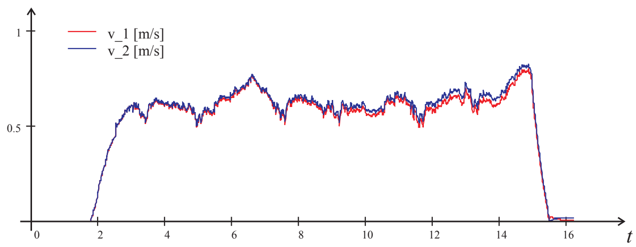

Figure 5. Smart phone velocity v_y in the direction of Y axis. The velocity v_y is calculated by: (a) algorithm 1 (v_1), which uses accY data that gives us the acceleration in the direction of movement and gyroZ data that gives us the information about smart phone orientation in the (x, y) plane, (b) algorithm 2 (v_2) uses the accX and accY data to directly calculate the velocity v_y.

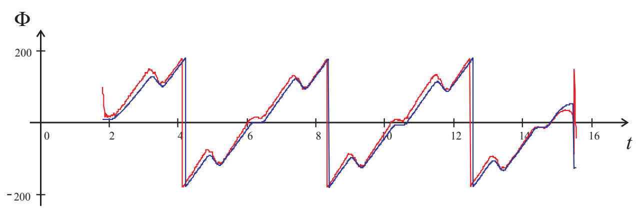

Figure 6. Rotation around Z axis. The smart phone is balanced in the (x, y) plane and the angle of the rotation is given in degrees (modulo 360). The blue curve corresponds to the algorithm 1 and the red curve to the algorithm 2.

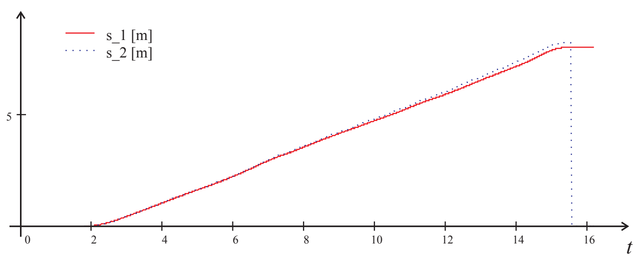

Figure 7. Path lengths calculated by both algorithms. We see that the path of algorithm 2 (s_1) is a little longer, what was expected as its velocities v_2 in Figure 5 are a bit above the velocities v_1 of algorithm 1.

5 Conclusion

By setting the correct offset values we can control tracking in a longer periods of time. Our experiments confirmed that accelerometer data are generally less accurate than gyroscope data, but still good enought for various motion-detection applications.

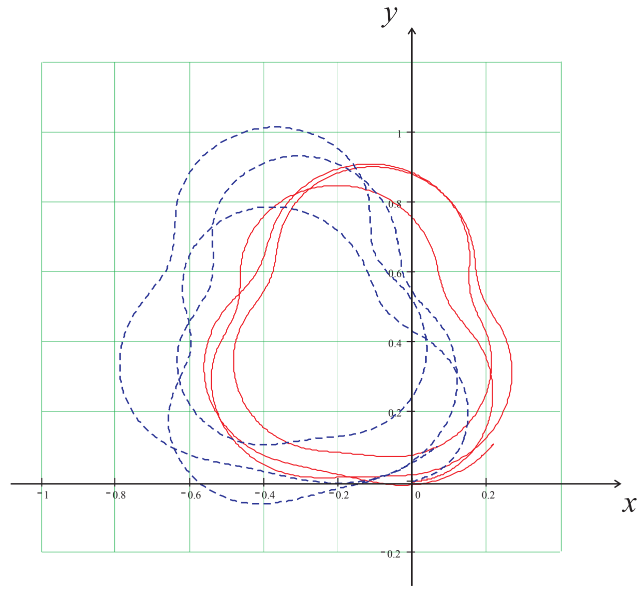

Figure 8. Calculated trajectory of the smart phone movement on the inner track from Figure 1. The solid curve shows the trajectory calculated by the algorithm 1 and the dashed curve shows he trajectory calculated by the algorithm 2. We notice that the results obtained by fusing the accelerometer and gyroscope data (algorithm 1) give much more faithful trajectory.

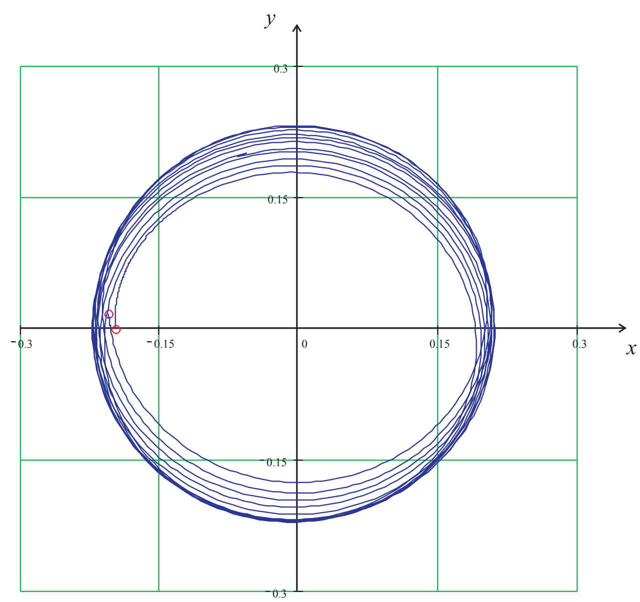

Figure 9. Calculated trajectory of the smart phone movement on the bicycle wheel 2. The trajectory is calculated by algorithm 1. We see that the iPhone makes circles with the radius of about $r = 22$ cm, what faithfully represents the conditions of the test.

6 References

[1] Motion sensing in the iPhone 4: MEMS accelerometer, http://www.memsjournal.com/2010/12/motion-sensing-in-the-iphone-4-mems-accelerometer.html

[2] STMicroelectronics, LIS331DLH MEMS digital output motion sensor ultra low-power high performance 3-axes "nano" accelerometer, http://www.st.com/web/catalog/sense power/FM89/SC444/PF218132

[3] STMicroelectronics, L3G4200D MEMS motion sensor ultra-stable three-axis digital output gyroscope http://www.st.com/web/catalog/sense power/FM89/SC1288/PF250373

[4] C. Barthold, P. Subbu, R. Dantu: Evaluation of Gyroscope-embeded Mobile Phones, Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, pg. 1532-1638,

[5] H.Graf, K.Jung: The Smartphone as a 3D Input Device, 2012 IEEE Second International Conference on Consumer Electronics, ICCE-Berlin, pg. 254-257,

[6] Xiaoji Niu et al, Using Inertial Sensors of iPhone 4 for Car Navigation, Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION , pg. 555-561.