Inertial Odometry on Handheld Smartphones

Arno Solin[1] Santiago Cortés[1] Esa Rahtu[2] Juho Kannala[1]

[1]Aalto University [2]Tampere University of Technology

21st International Conference on Information Fusion

July 12, 2018

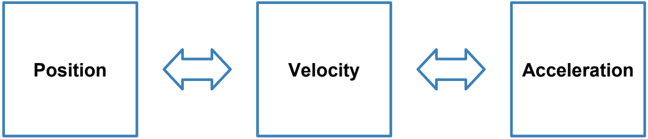
Introduction

- Phones have accelerometers and gyroscopes
- Should in theory enable inertial navigation
- Cheap and small sensors
- Low quality data
  ➔ inertial navigation not possible

! We demonstrate that it can be done

Inertial navigation: How it should work

- Velocity is the integral of acceleration.
- Position is the integral of velocity.
- We can observe acceleration and angular velocity in the mobile phone.



Inertial navigation: Why it does not work

- All inertial navigation systems suffer from integration drift.
- Small errors in the measurement of acceleration and angular velocity...
- Progressively larger errors in velocity...
- Even greater errors in position.

- The dominating component in acceleration is gravity.
- Even slight error in orientation makes the gravity 'leak'.

- The sequential nature of the problem makes the errors accumulate.

Inertial navigation: How to make it work

- Input: accelerometer data $\mathbf{a}_k$ and gyroscope data ω_k.
- Accelerometer and gyroscope biases part of the state:

$$\tilde{\mathbf{a}}_k = \mathbf{T}^{a}_k \, \mathbf{a}_k - \mathbf{b}^{a}_k \qquad \tilde{\omega}_k = \omega_k - \mathbf{b}^{\omega}_k$$

- Dynamical model:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{q}_k \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \Delta t_k \\ \mathbf{v}_{k-1} + [\mathbf{q}_k(\tilde{\mathbf{a}}_k + \varepsilon^{a}_k)\mathbf{q}^{\star}_k - \mathbf{g}]\Delta t_k \\ \Omega[(\tilde{\omega}_k + \varepsilon^{\omega}_k)\Delta t_k]\mathbf{q}_{k-1} \end{pmatrix}$$

  for position $\mathbf{p}_k$, velocity $\mathbf{v}_k$, and orientations $\mathbf{q}_k$ at time t_k.
- Inference by an Extended Kalman filter / smoother

Inertial navigation: How to make it work

- Additional constraints (observations) are required
- This framework can use
  - Zero-velocity updates (ZUPTs)
  - Position fixes
  - Loop-closures
  - Barometric air pressure for relative height
- A pseudo-measurement keeping the velocity component from exploding
- Sensor timing info
- A matter of learning the biases

How does this compare to previous approaches?

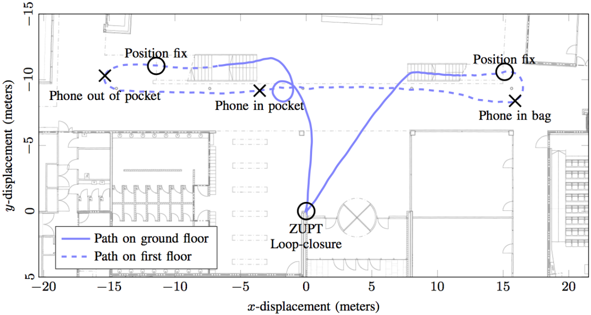
On mobile devices PDR typically based on:

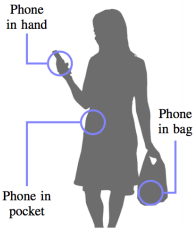
- Movement detection
- Step and heading systems
- Visual features
- All of these are limited in some way

Example studies

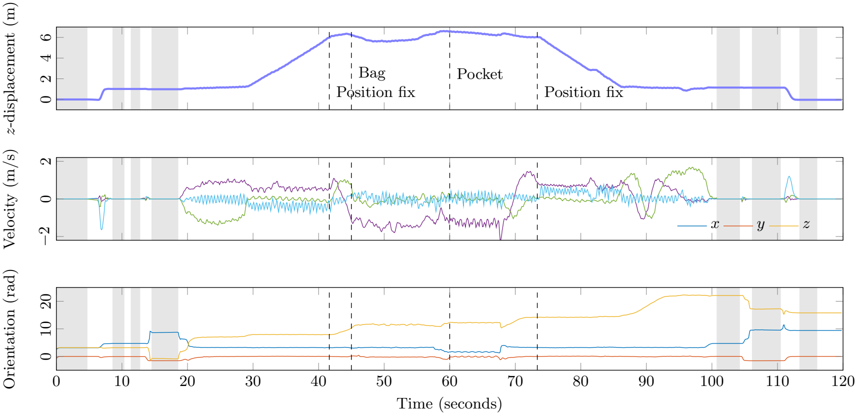
- Equipment used:
  - Off-the-shelf iPhone 6
- Sensors:
  - Built-in gyroscope and accelerometer
  - Sampling rate: 100 Hz
- Computations:
  - Off-line
    (runnable on device hardware)

Example: Conventional PDR

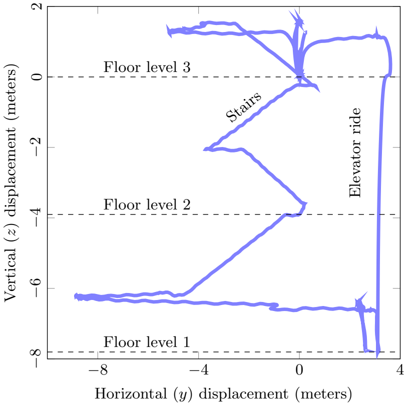




Example: Conventional PDR

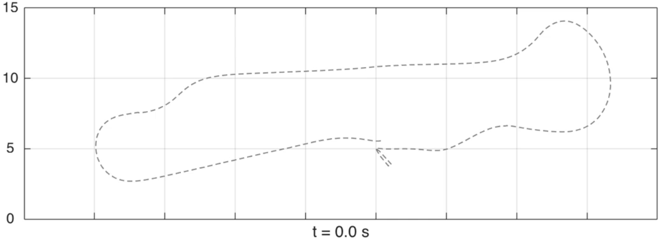


Example: Conventional PDR



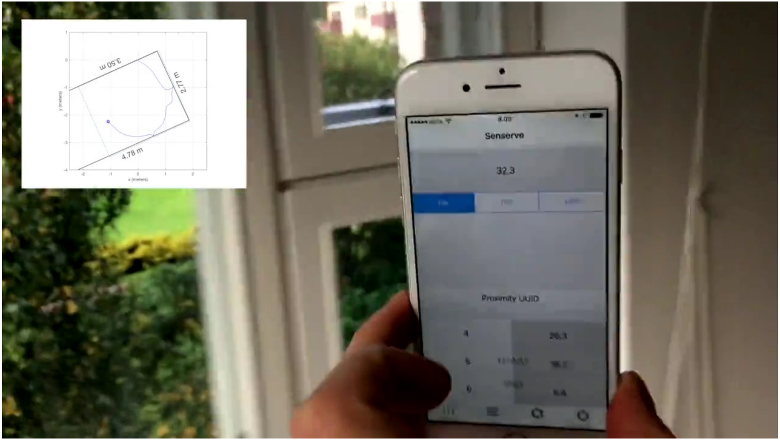
Example: General dead-reckoning

Example: General dead-reckoning



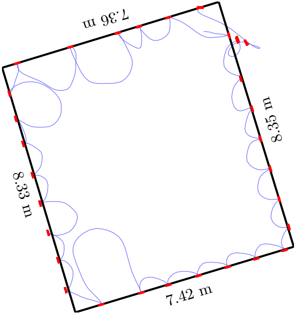
The video is available on YouTube:
`https://youtu.be/L-E9fNsrvII`

Example: Inertial measurements

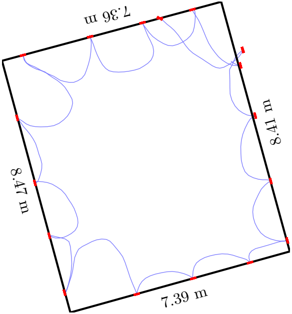


The video is available on YouTube:
`https://youtu.be/L-E9fNsrvII`

Example: Inertial measurements



(a) Measurement #1



(b) Measurement #2

Recap

- Inertial navigation on a smartphone is possible
- The key is learning the sensor transformations (biases)
- Not possible without measurements (consraints)
  - ZUPTs
  - pseudo-speed
- Future work towards relaxing these further

- Homepage:
  http://arno.solin.fi
- Twitter:
  @arnosolin