Short-term Motion Tracking Using Inexpensive Sensors

Filip Matzner, Roman Barták

Introduction

- **Sensors** became extremely small, lightweight, low-power and cheap.
- In **theory**, having perfect inertial sensors, it is possible to track the full **3D motion**.
- Is it possible in **practice** with cheap low-end sensors present in a smartphone?

Outline

- Sensor Description
  - What sensors do we need and what data they produce.
- Sensor Fusion
  - Fusion of the sensor data into the position.
- Experiment with a Smartphone
  - Does it work? Try yourself!

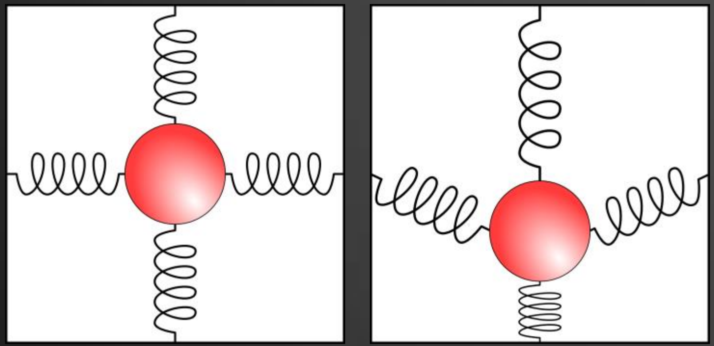
Sensors

- **MEMS** (micro-electro-mechanical-systems)
- Usual combination present in smartphones is:
  - accelerometer
  - gyroscope
  - magnetometer

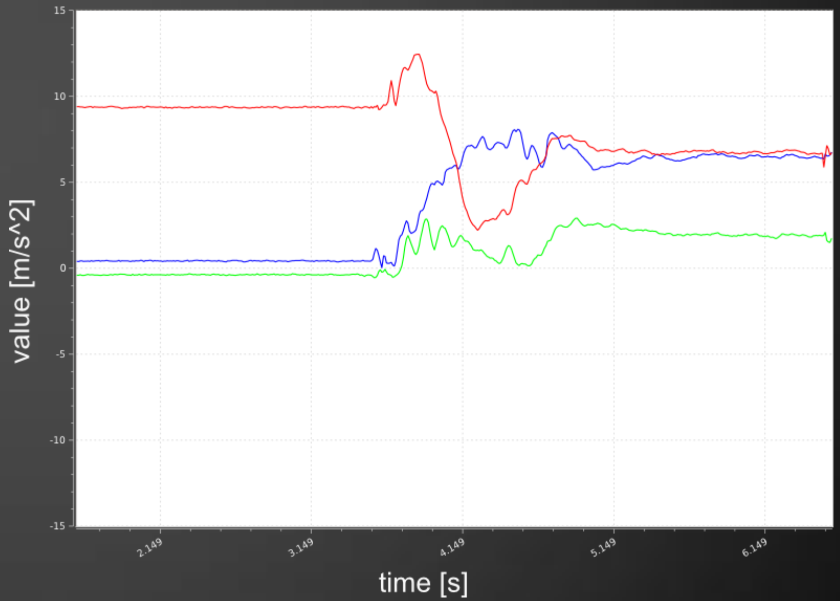
Sensors - accelerometer

- Measures **acceleration** of the device in three coordinate axes [ms^{-2}].

2D accelerometer concept upward acceleration

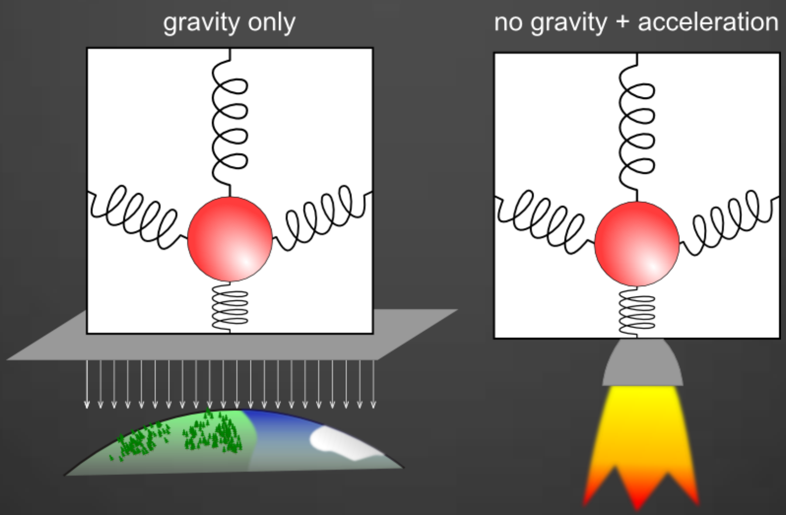


raw 3-axis accelerometer data



Sensors - accelerometer (gravity)

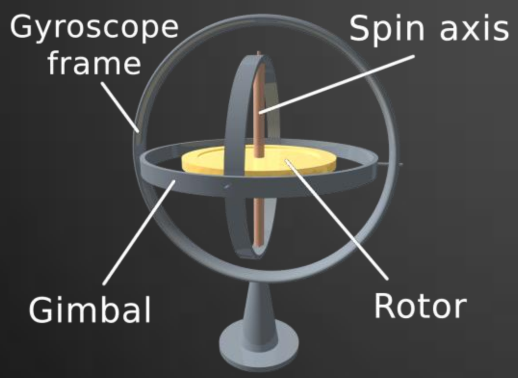
- Every accelerometer is affected by a permanent **gravity force.**
- The gravity force cannot be distinguished from upward acceleration.



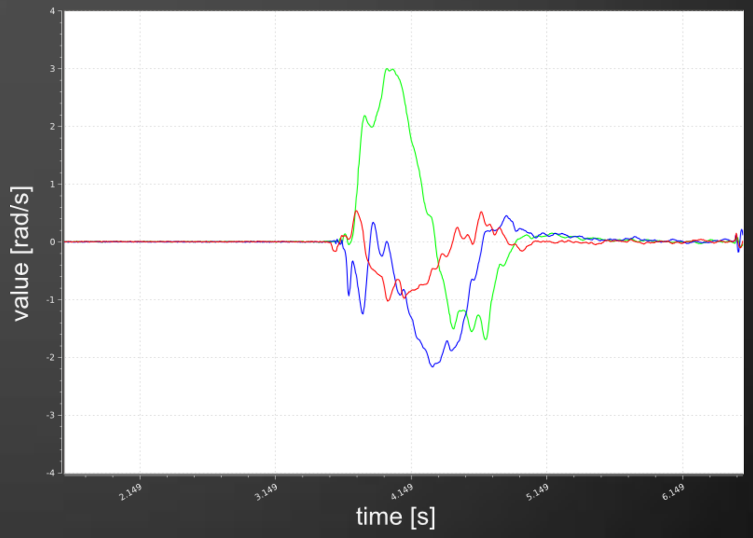
Sensors - gyroscope

- Historical gyroscopes were able to produce **absolute orientation** in space.
- MEMS gyroscopes only produce **angular speed** [rad/s] for each coordinate axis.

historical gyroscope concept

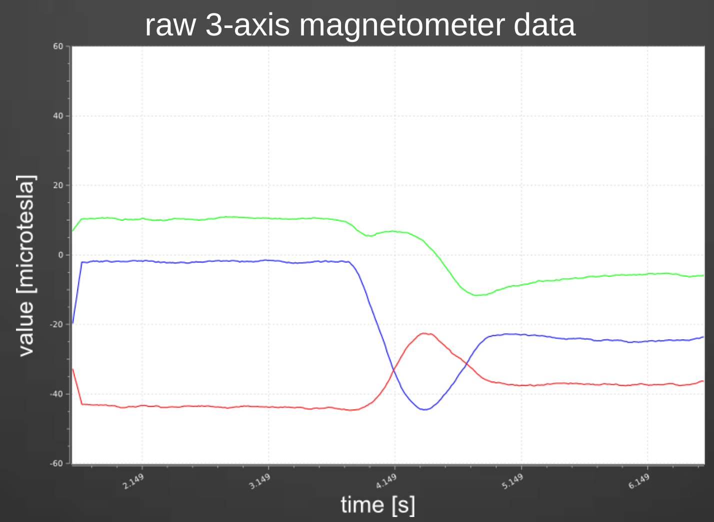


raw 3-axis MEMS gyroscope data



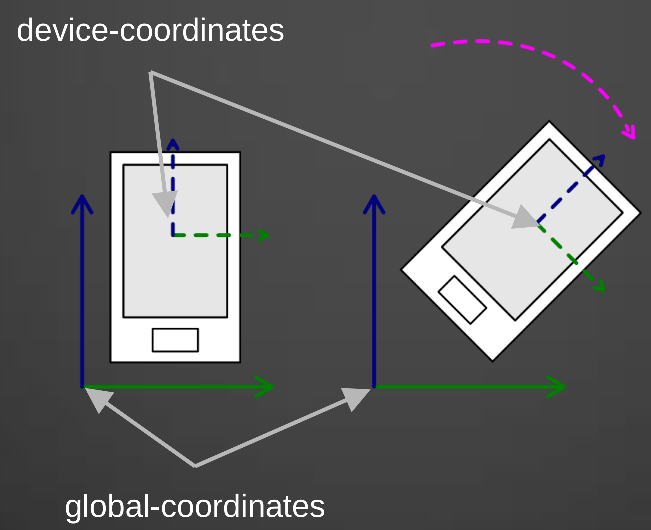
Sensors - magnetometer

- Measures **magnetic field** vector, i.e., magnetic field strength for each coordinate axis.
- Sensitive to an interference caused by WiFi, electrical wires, etc.



Sensor Fusion - coordinate systems

- We will be working with two coordinate systems.
  - **global-coordinates**: the system the device was in at the beginning
  - **device-coordinates**: the system fixed to the device



Sensor Fusion - problem specification

- We have the following data in device-coordinates:
  - **acc(t)** = accelerometer vector in time t
  - **gyro(t)** = gyroscope vector in time t
  - **mag(t)** = magnetic field vector in time t
- We want to fuse the data to get the position of the device in time t in global-coordinates.
  - **pos(t)** = position vector in time t

Sensor Fusion - theoretical model 1

The device **does not rotate** during the entire motion.

- First some basic definitions from physics:

$$
\begin{aligned}
\mathrm{linacc} : \mathbb{R} \to \mathbb{R}^3 &\text{ acceleration in time } t \text{ without gravity} \\
\mathrm{vel} : \mathbb{R} \to \mathbb{R}^3 &\text{ velocity in time } t \\
\mathrm{pos} : \mathbb{R} \to \mathbb{R}^3 &\text{ position in time } t \\
\mathrm{vel}(t) &= \frac{\partial \mathrm{pos}(t)}{\partial t} \\
\mathrm{linacc}(t) &= \frac{\partial \mathrm{vel}(t)}{\partial t}
\end{aligned}
$$

  - But we only have the accelerometer data!
    - We will use the definition the other way around:

$$
\begin{aligned}
\mathrm{vel}(t) &= \int_0^t \mathrm{linacc}(u) \mathrm{d}u \\
\mathrm{pos}(t) &= \int_0^t \mathrm{vel}(u) \mathrm{d}u
\end{aligned}
$$

Sensor Fusion - theoretical model 1

The device **does not rotate** during the entire motion.

- The accelerometer vector also measures gravity, which has to be subtracted:

$$\mathrm{linacc}(t) = \mathrm{acc}(t) - \mathrm{acc}(0)$$

- Now we have the formula for the position of the device as long as it does not rotate.

$$\mathrm{pos}(t) = \int_0^t \int_0^t \mathrm{acc}(u) - \mathrm{acc}(0) \mathrm{d}^2 u$$

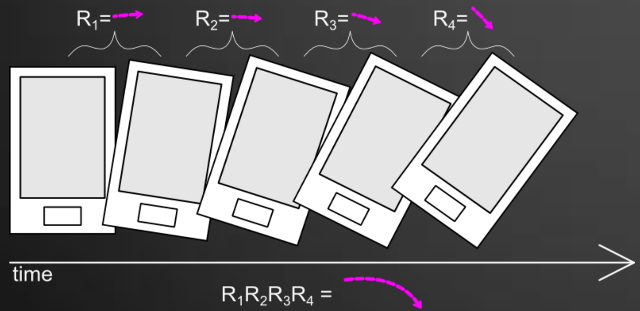
Sensor Fusion - theoretical model 2

The device **rotates** during the motion.

- If we allow **rotation**, the sensor data from different time can be relative to different coordinate systems.
- To **convert** the data from device-coordinates to global-coordinates, we need to know the **orientation** of the device.
- The orientation will be represented by a 3x3 **rotation matrix**.

Sensor Fusion - Orientation

- The orientation will be a **rotation matrix** representing the rotation of the device-coordinate system relative to the global-coordinate system.
- The matrix will be created by **sampling** the data from the gyroscope into rotation matrices and **multiplying** the matrices together.



$$\text{ori} : \mathbb{R} \to \mathbb{R}^{3\times 3}$$

$$\text{ori}(t) := \lim_{h\to 0} \prod_{i=1}^{\lfloor t/h \rfloor} \Delta\text{rot}_{\text{mat}}((i-1)h, ih)$$

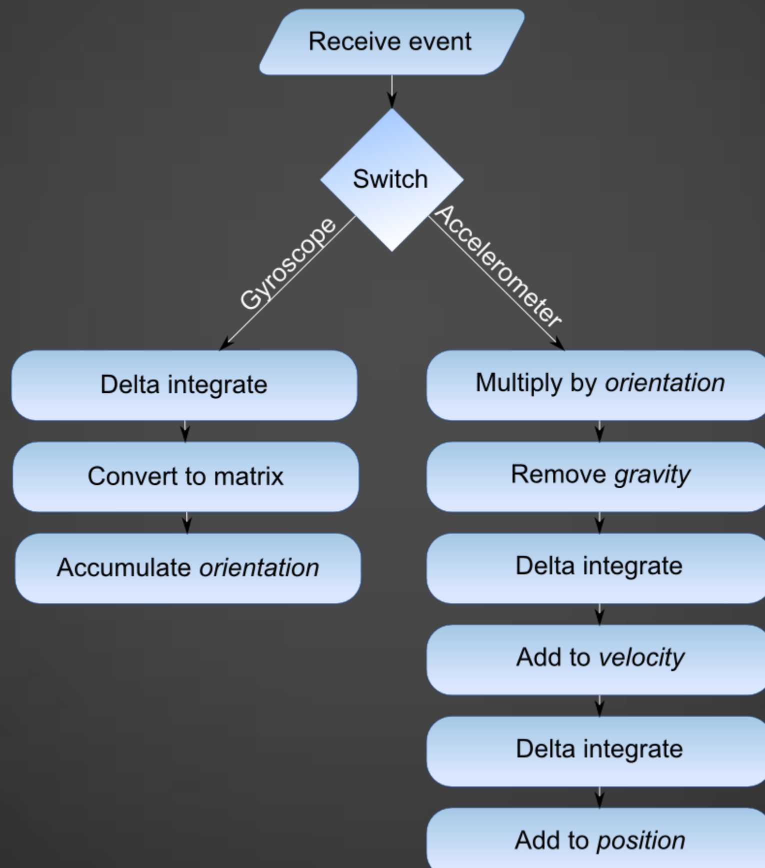
Sensor Fusion - the final formula

- Putting the orientation formula and the simple position formula together gives us the final formula for the position including the rotation of the device.

$$\mathrm{pos} : \mathbb{R} \to \mathbb{R}^3$$

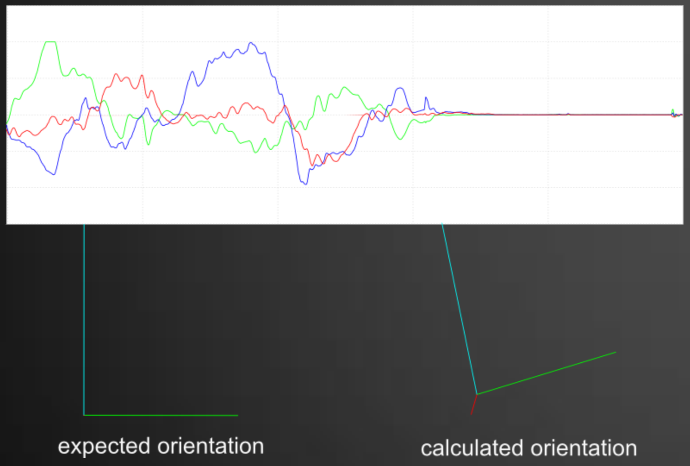
$$\mathrm{pos}(t) := \int_0^t \int_0^t \mathrm{ori}(u)\mathrm{acc}(u) - \mathrm{acc}(0)\ \mathrm{d}^2 u$$

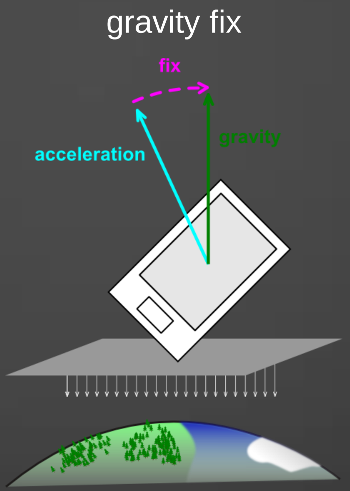
Sensor Fusion - algorithm

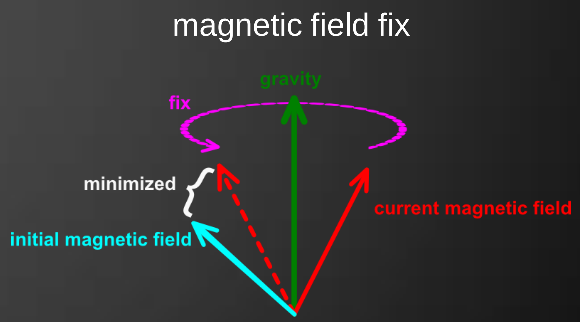


Automated Stabilization

- The gyroscope **bias** introduces a drift, which can be **compensated** by gravity and magnetic field.

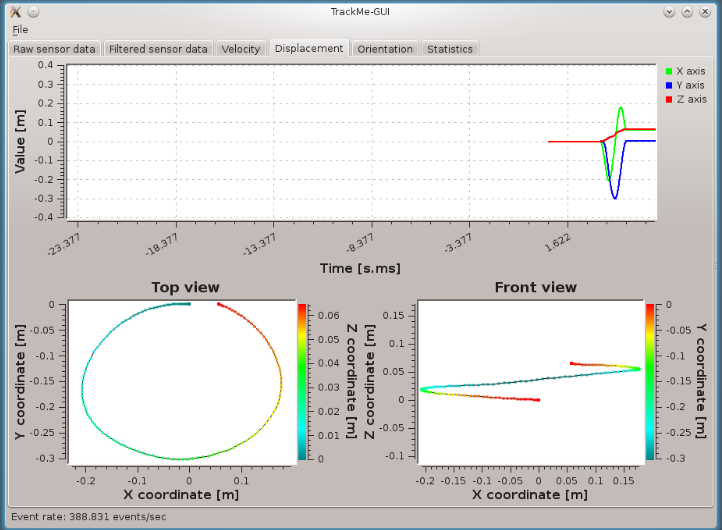






Experimental Evaluation

- We have developed a software and performed a series of experiments.



Conclusion

- The cheap sensors are not accurate enough to track long motions.
- If the motion can be separated into multiple short intervals, the automated stabilization can prolong reliability.
- A model method, such as **Kalman filter**, might improve the results and reduce noise.