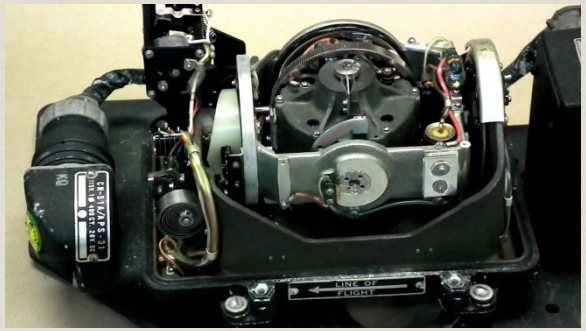
Inertial Navigation and Various Applications of Inertial Data

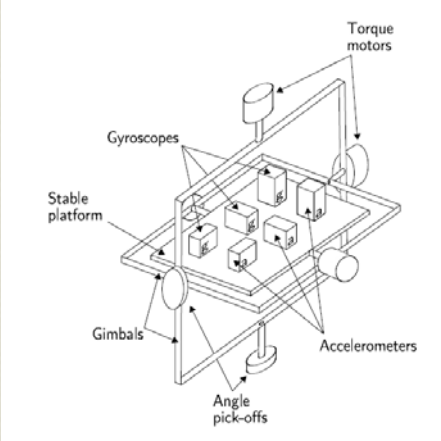
Yongcai Wang

9 November 2016

Types of Gyroscope

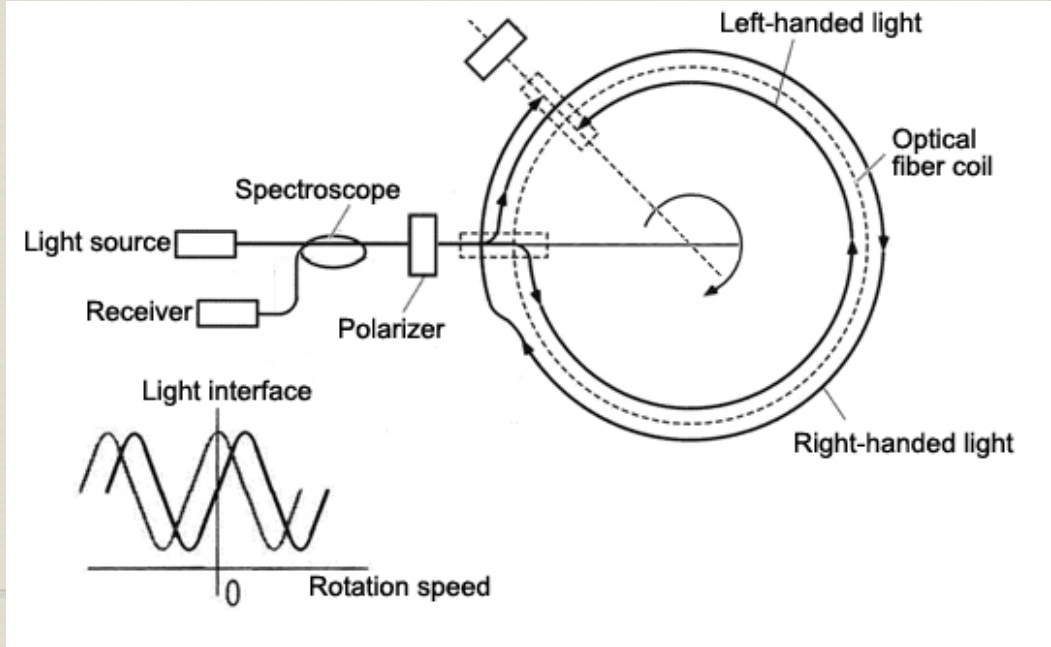
- Mechanical Gyroscope



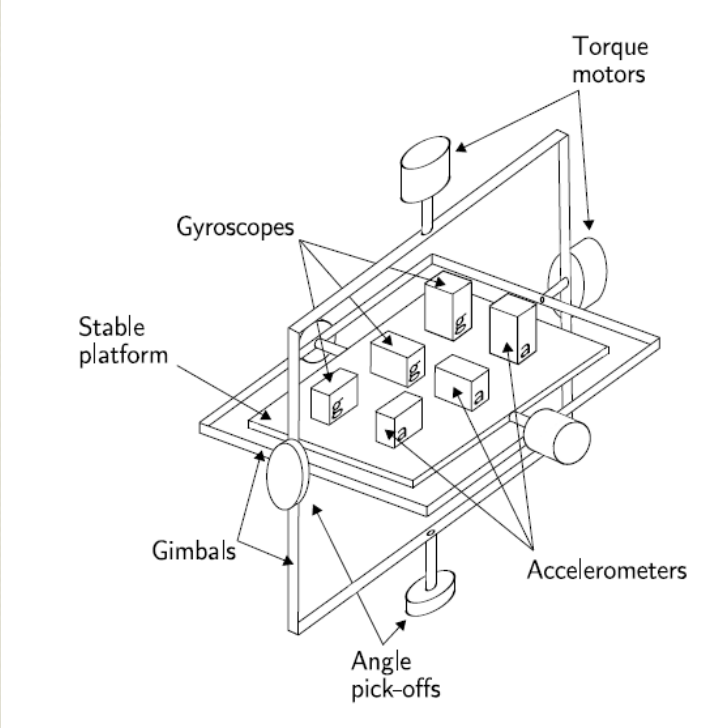


- Laser Gyroscope

Sagnac Effect



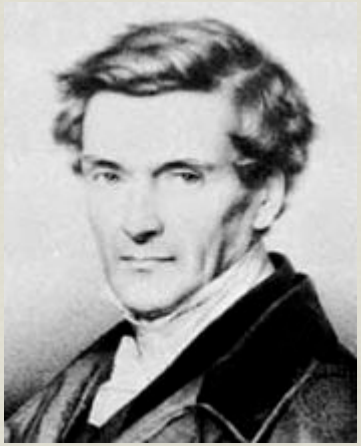
Stable Platform IMU and Strapdown IMU



In Stable platform, the inertial sensors are mounted on a platform which is isolated from any external rotational motion, **returning readings in global frame.**

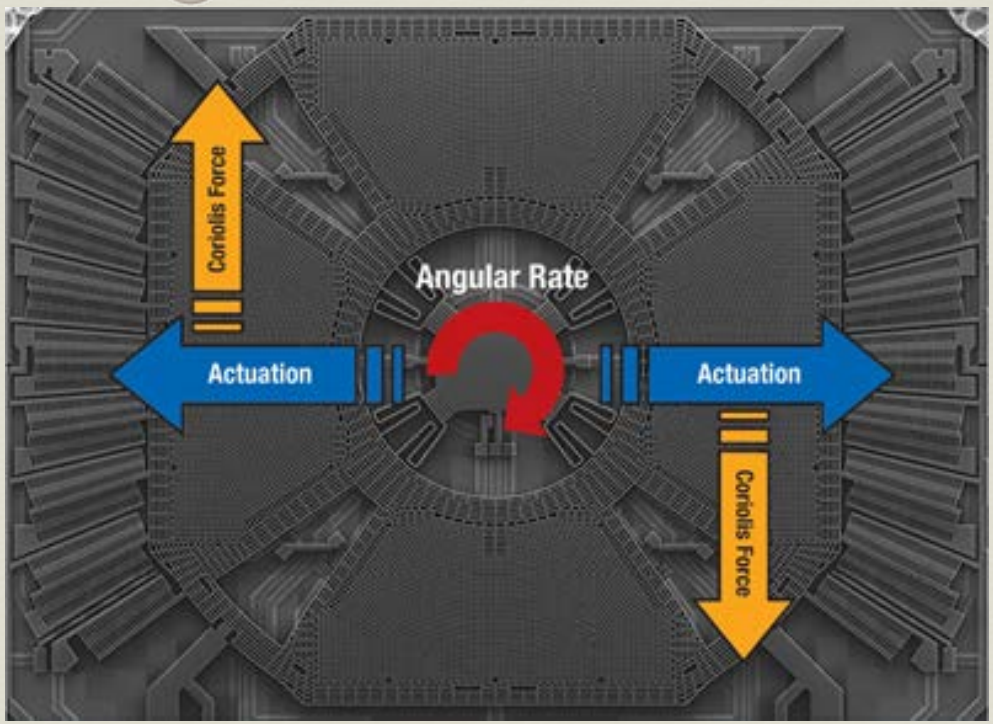
In strapdown systems the inertial sensors are mounted rigidly onto the device, **returning readings in body frame.**

Types of Gyroscopes: MEMS



Coriolis force

F=2mv*w



Accuracy Comparison

	GG1320AN (Laser)	MPU6050 (MEMS)
Size	88*88*45	5*5*5
Power	1.6Watt	5V, 3.6mA
Angular random walk	**0.0035** $^{o}/\sqrt{h}$	<0.2 $^{o}/\sqrt{h}$
Bias stability	**0.0035** $^{o}/\sqrt{h}$	<70 $^{o}/\sqrt{h}$
Price	>5000	20

Inertial Navigation System

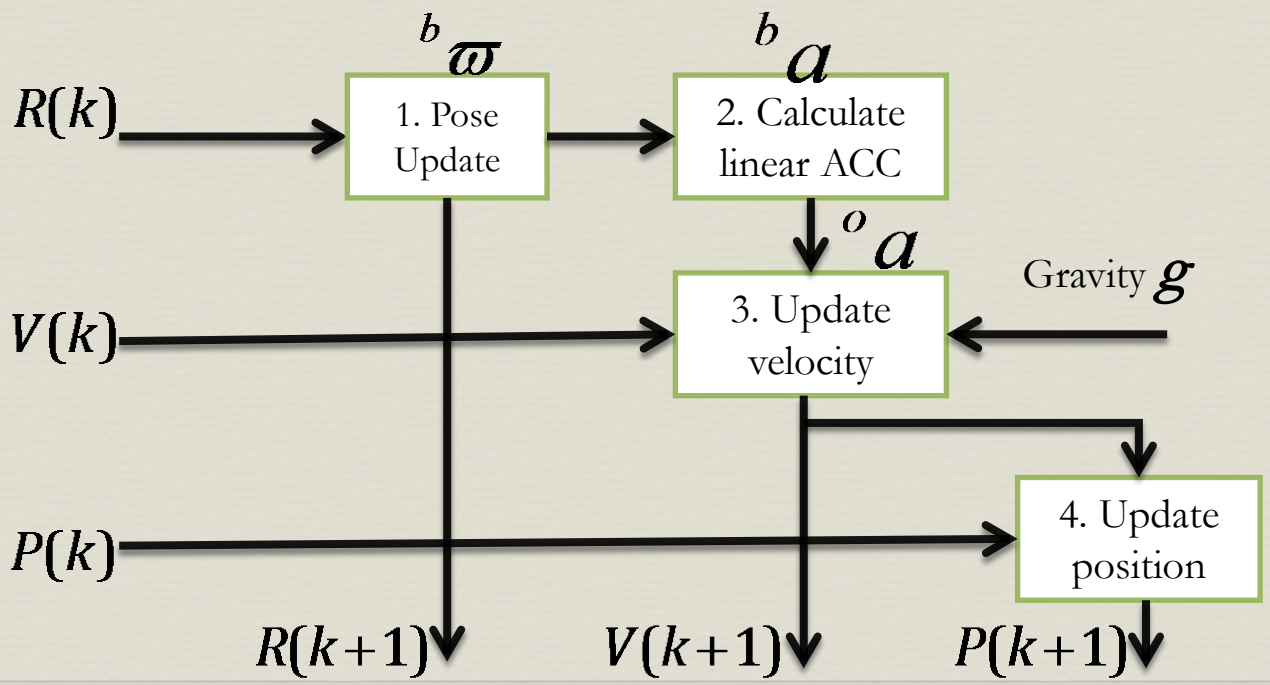
Inertial Navigation Systems

- Gyroscope provides angle velocity in body frame

$$\varpi=(\omega_x,\omega_y,\omega_z)$$

- Accelerometer provides acceleration in body frame

$$a=(a_x,a_y,a_z)$$



Outline

- Pose update method
- Velocity and position update
- Error correction
- Step segmentation
- Pedestrian dead reckoning

Coordinate Frame Rotating

$$R(t)R^T(t) = 1$$

$$\Rightarrow \dot{R}(t)R^T(t) + R(t)\dot{R}^T(t) = 0$$

$$S(t) = \dot{R}(t)R^T(t)$$

$$\Rightarrow S(t) = -S^T(t) \Rightarrow \dot{R}(t) = S(t)R(t)$$

Skew Symmetric matrix

- Angular velocity: $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$

The derivative of a time-varying rotation matrix:

$$\dot{R}(t) = S(\varpi)R(t)$$

$$S(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{pmatrix}$$

Coordinate Frame Rotating

- **By first order Taylor explanation**

$$
\begin{aligned}
R(t+\delta_t) &\approx \delta_t \dot{R}(t) + R(t) \\
&\approx \delta_t S(\varpi) R(t) + R(t) \\
&\approx \left(\delta_t S(\varpi) + I_{3\times3}\right) R(t)
\end{aligned}
$$

How the rotation matrix change according to the angular rate

- **Derivate the angular change from the rotation matrix**

$$\delta_t S(\boldsymbol{\omega}) = R(t+\delta_t) R(t)^T - I_{3\times3}$$

$$
\begin{aligned}
\delta_\Theta &= \delta_t \varpi \\
&= vex\left(\delta_t S(\varpi)\right) \\
&= vex\left(R(t+\delta_t) R(t)^T - I_{3\times3}\right)
\end{aligned}
$$

How to derive the angular change from the rotation matrix

Continuous Rotation

- **Infinitesimal Rotating is Commutive**

$\mathbf{R_{xyz}}$=rotx(0.001)*roty(0.002)*rotz(0.003)

$\mathbf{R_{yxz}}$=roty(0.002)*rotx(0.002)*rotz(0.003)

$\mathbf{R_{xyz}=R_{yxz}}$

- **Recovery of the infinitesimal rotation angle** δ_Θ

vex($\mathbf{R_{xyz}}$)=vex($\mathbf{R_{yxz}}$) =(0.001, 0.002, 0.003)

- **Difference between two infinitesimally poses**

$$\delta = \Delta\left(\xi_0, \xi_1\right) = \left(\delta_d, \delta_\Theta\right)$$

δ_d Incremental displacement

δ_Θ Incremental rotation

1. Pose update (1)

- **1. Pose update by first order Taylor approximation:**

$$R(k+1) = \delta_t S\left(\varpi\right) R(k) + R(k)$$

- **2. Pose update by higher order approximation:**

$$\dot{R}(t) = S\left(\varpi\right) R(t)$$

$$\Rightarrow R(t) = R(0) \cdot \exp\left(\int_0^t \Omega(t) dt\right)$$

Let $\int_{t-\delta_t}^{t} \Omega(t) dt = \mathrm{B}_t$ When δt is small:

$$\mathrm{B}_t = \begin{pmatrix} 0 & -\varpi_t^z \delta t & \varpi_t^y \delta t \\ \varpi_t^z \delta t & 0 & -\varpi_t^x \delta t \\ -\varpi_t^y \delta t & \varpi_t^x \delta t & 0 \end{pmatrix}$$

$$\Rightarrow R(t) = R(t - \delta t) \cdot \exp\left(\mathrm{B}_t\right)$$

1. Pose update (2)

$$R(t) = R(t-\delta t)\cdot \exp\left(\mathrm{B}_t\right)$$

$$= R(t-\delta t)\left(I + \mathrm{B}_t + \frac{\mathrm{B}_t^2}{2!} + \frac{\mathrm{B}_t^3}{3!} + \frac{\mathrm{B}_t^4}{4!} + \cdots \right)$$

Let $\sigma = |\varpi_t^b \delta t|$, it is easy to verify that $\mathrm{B}_t^2 = -\sigma^2$

$$R(t) = R(t-\delta t)\left(I + \mathrm{B}_t + \frac{\mathrm{B}_t^2}{2!} + \frac{\mathrm{B}_t^3}{3!} + \frac{\mathrm{B}_t^4}{4!} + \ldots \right)$$

$$= R(t-\delta t)\left(I + \mathrm{B}_t + \frac{\mathrm{B}_t^2}{2!} - \frac{\sigma^2 \mathrm{B}_t}{3!} - \frac{\sigma^2 \mathrm{B}_t^2}{4!} + \ldots \right)$$

$$= R(t-\delta t)\left(I + \left(1 - \frac{\sigma^2}{3!} + \frac{\sigma^4}{5!} + \cdots \right)\mathrm{B}_t + \left(\frac{1}{2!} - \frac{\sigma^2}{4!} + \frac{\sigma^4}{5!} + \cdots \right)\mathrm{B}_t^2 \right)$$

$$= R(t-\delta t)\left(I + \frac{\sin\sigma}{\sigma}\mathrm{B}_t + \frac{1-\cos\sigma}{\sigma^2}\mathrm{B}_t^2 \right)$$

Orthogonarize and Normalization

- R(t) may becomes not orthonormal

  If the added term is small, the result will be close to orthonormal and we can *straighten it up*

$$c_3' = c_3$$

$$c_1' = c_2 \times c_3'$$

$$c_2' = c_1' \times c_3'$$

$$c_i'' = \frac{c_i'}{\left| c_i' \right|}$$

Normalization should be carried out after each step of integration

2. Pose Update by Quaternions

- It become easier when using unit-quaternions.
- The derivative of q:

$$\dot{q} = \frac{1}{2} q(w) q, \text{ where } q(w) = 0, < w >$$

- Integration of quaternion rate:

$$q(k+1) = \delta_t \dot{q} + q(k)$$

- Normalization

$$q'(k+1) = \frac{q(k+1)}{\left| q(k+1) \right|}$$

3. Velocity and Location Update

- Convert the acceleration to global frame

$$a_t^g = R(t) a_t^b$$

- Exclude the gravity

$$a_t^g = a_t^g - g^g$$

- Double integral to calculate velocity and position

$$v_t^g = v_{t-\delta t}^g + \delta t \cdot a_t^g$$

$$p_t^g = p_{t-\delta t}^g + \delta t \cdot v_t^g$$

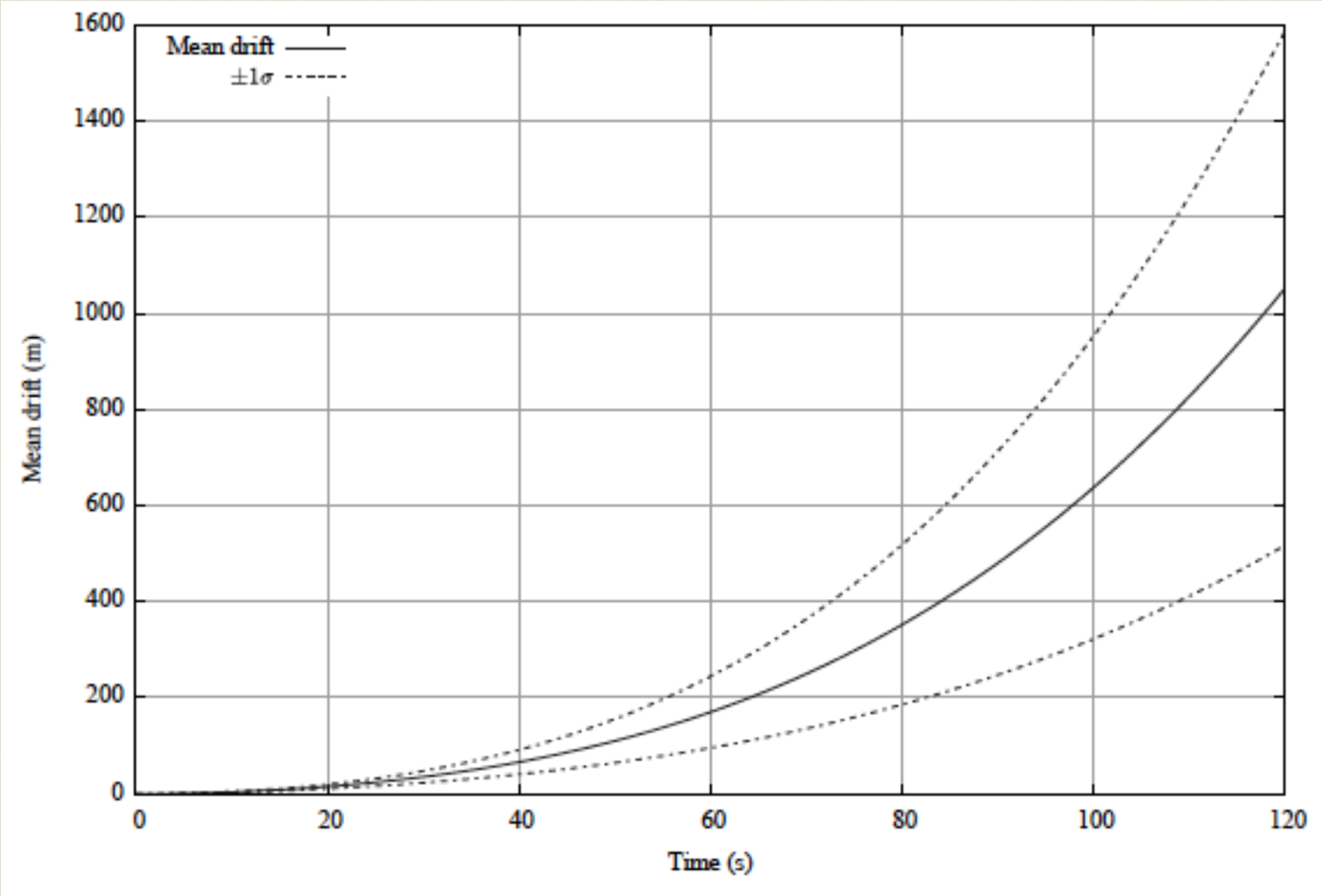
MEMS gyroscope error sources

Error type	Description	Result of single integration
Bias	A constant bias offset	A linearly growing error in angle
Random noise	Random noise from a variety of sources (usually specified in terms of its root PSD)	An angle random walk whose standard deviation grows proportional to the square root of time
Calibration	Deterministic errors in scale factors, alignments and gyro linearities	Orientation drift proportional to the rate and duration of motion
Temperature effects	Temperature dependent residual bias	Any residual bias is integrated into the orientation, causing an orientation error which grows linearly with time
Bias instability	Bias fluctuations, usually modelled as a bias random walk	A second-order random walk

$$e_t = N(0, N(0,t)) + N(0,t) + c + N(0,Tt)$$

bias Random noise calibration Temperature effects

INS Example


Mean drift
±1σ
Mean drift (m)
Time (s)
0
200
400
600
800
1000
1200
1400
1600
0
20
40
60
80
100
120

INS Error Correction

- **Using external measurements to correct the states of IMU.**
  - **Zero Velocity Update (ZUPT)**
  - **GPS, Other locating system**
- **Using a Kalman Filter to correct an INS based on external measurements.**
  - **Tracking full state of INS (openshoe)**
  - **Tracking state error of INS**

INS Error Correction

- **The true state of an IMU at time *t***

$$(R_t, v_t, p_t)$$

- **The solution calculated by INS**

$$(\hat{R}_t, \hat{v}_t, \hat{p}_t)$$

- **The error state**

$$\delta x_t = \left(\delta\Phi_t^T, \delta v_t^T, \delta p_t^T\right)^T \quad \text{in which} \quad \delta\Phi_t = \left(\delta\phi_t^{gx}, \delta\phi_t^{gy}, \delta\phi_t^{gz}\right)$$

- **Relation of Error state and IMU State:**

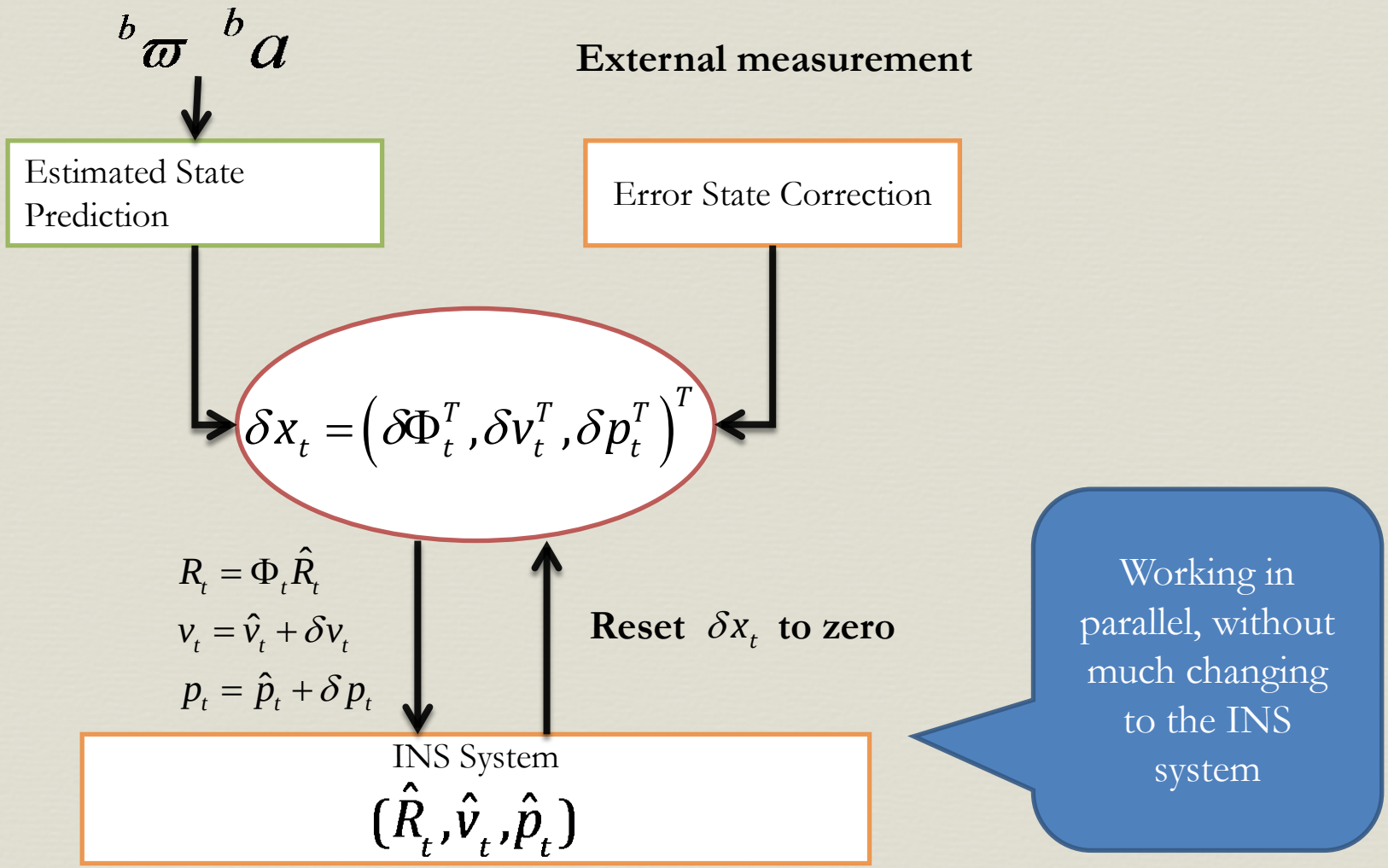
$$R_t = \Phi_t \hat{R}_t$$

$$v_t = \hat{v}_t + \delta v_t$$

$$p_t = \hat{p}_t + \delta p_t$$

$$\Phi_t = \begin{pmatrix} 1 & -\delta\phi_t^{gz} & \delta\phi_t^{gy} \\ \delta\phi_t^{gz} & 1 & -\delta\phi_t^{gx} \\ -\delta\phi_t^{gy} & \delta\phi_t^{gx} & 1 \end{pmatrix}$$

INS Error Correction by KF



Error State Prediction

- **State transition matrix**

$$F_t = \begin{pmatrix} I & 0 & 0 \\ \delta_t S(a_t^g) & I & 0 \\ 0 & \delta_t I & I \end{pmatrix}$$

where

$$S(a_t^g) = \begin{pmatrix} 0 & -a_t^{gz} & a_t^{gz} \\ a_t^{gz} & 0 & -a_t^{gx} \\ -a_t^{gy} & a_t^{gx} & 0 \end{pmatrix}$$

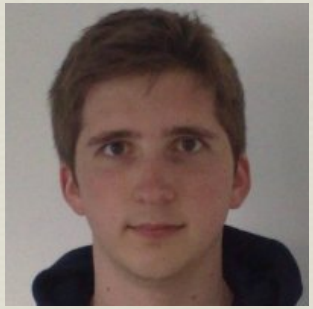
- **Error State Prediction**

$$\delta \hat{x}_t^- = F_t \delta \hat{x}_{t-\delta t}^-$$

$$P_t^- = F_t P_{t-\delta t} F_t^T + Q_t$$

Eric Foxlin

Founder of intersense



Oliver Woodman

In Google Android group

[1] Pedestrian tracking with shoe-mounted inertial sensors, E. Foxlin 2005, IEEE computer graphics and applications.
[2] Pedestrian localisation for indoor environments, Oliver Woodman, 2010, PhD thesis, Cambridge university

Error State Prediction

$$Q_t = \begin{pmatrix} T_t & 0 & 0 \\ 0 & U_t & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The noise item in state prediction

$$T_t = \delta t^2 \left(\hat{R}_t^-\right)^T W_t \left(\hat{R}_t^-\right)$$

$$W_t = \begin{pmatrix} (\sigma_w^{bx})^2 & 0 & 0 \\ 0 & (\sigma_w^{by})^2 & 0 \\ 0 & 0 & (\sigma_w^{bz})^2 \end{pmatrix}$$

Representing noises of gyroscope

$$U_t = \delta t^2 \left(\hat{R}_t^-\right)^T A_t \left(\hat{R}_t^-\right)$$

$$A_t = \begin{pmatrix} (\sigma_a^{bx})^2 & 0 & 0 \\ 0 & (\sigma_a^{by})^2 & 0 \\ 0 & 0 & (\sigma_a^{bz})^2 \end{pmatrix}$$

Representing noises of accelerometer

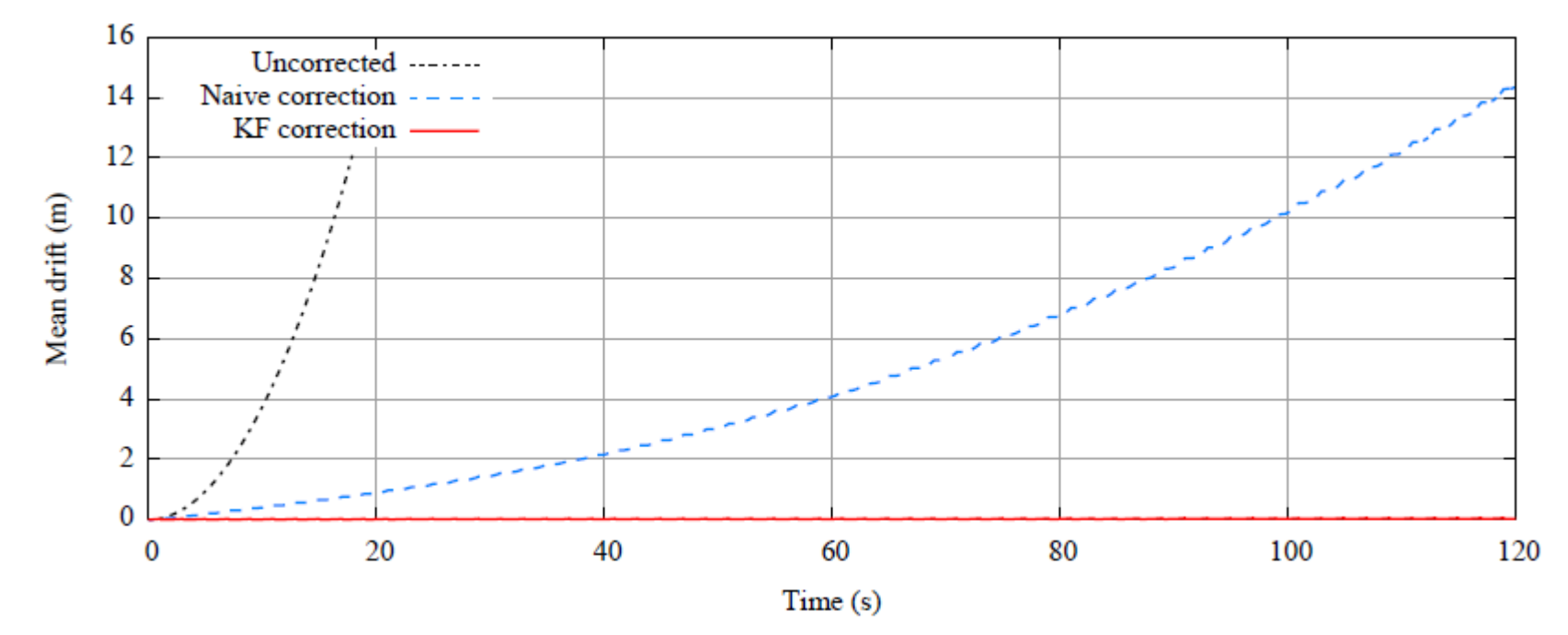
Error State Correction

- **External measurement**
  - ZUPT
  - Ultrasound…

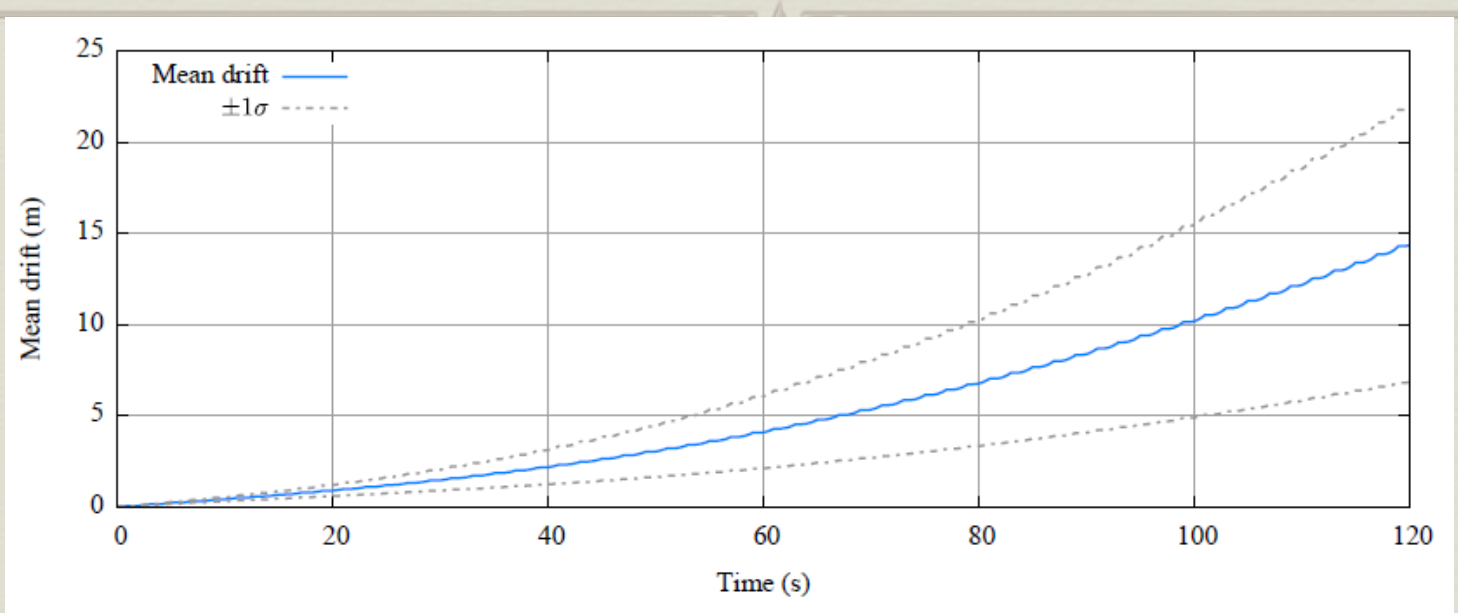
$$z_t = h_t(x_t) + v_t$$

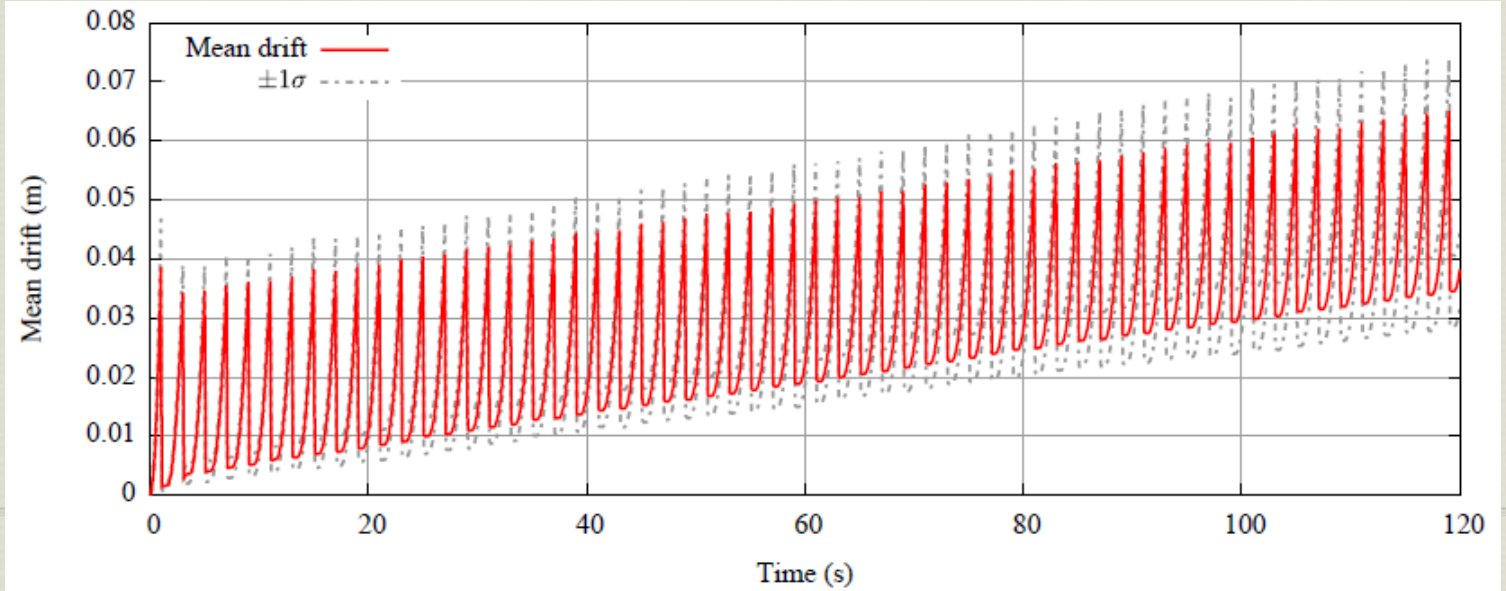
In a simple test by keeping the IMU on table

$$h_t(x_t) = v_t$$



KF-based correction vs. naïve correction

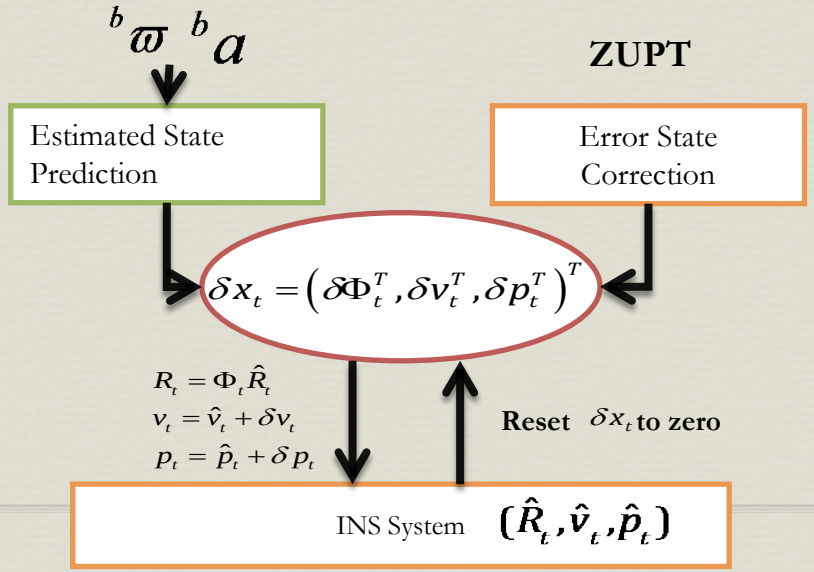


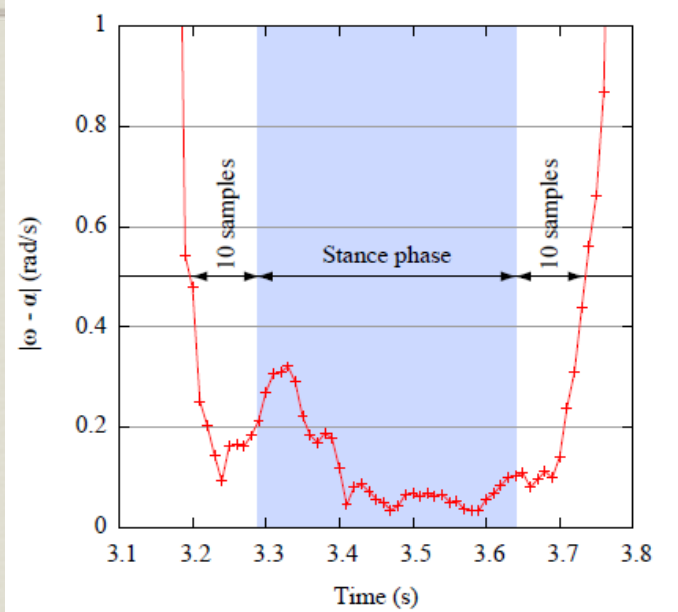


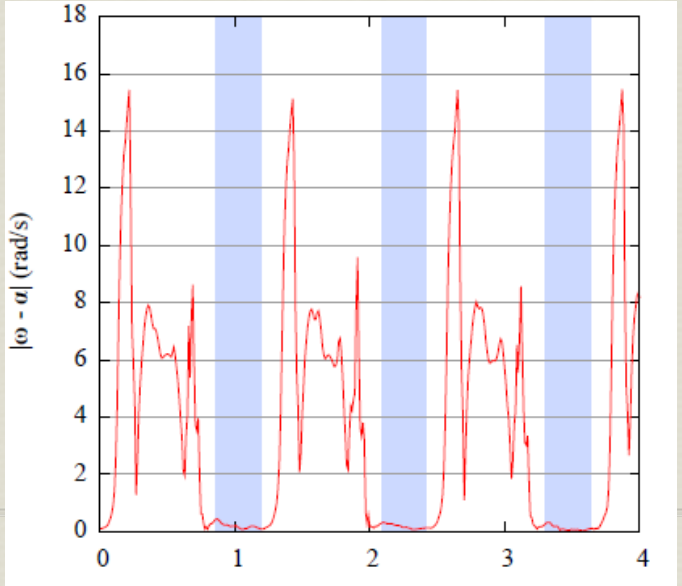
Pedestrian Dead Reckoning

- Zero Velocity detection: ZUPT

$$|\omega_t^b - \hat{\alpha}_t| < 0.5\,\mathrm{rad/s}\,.$$







Step Segmentation

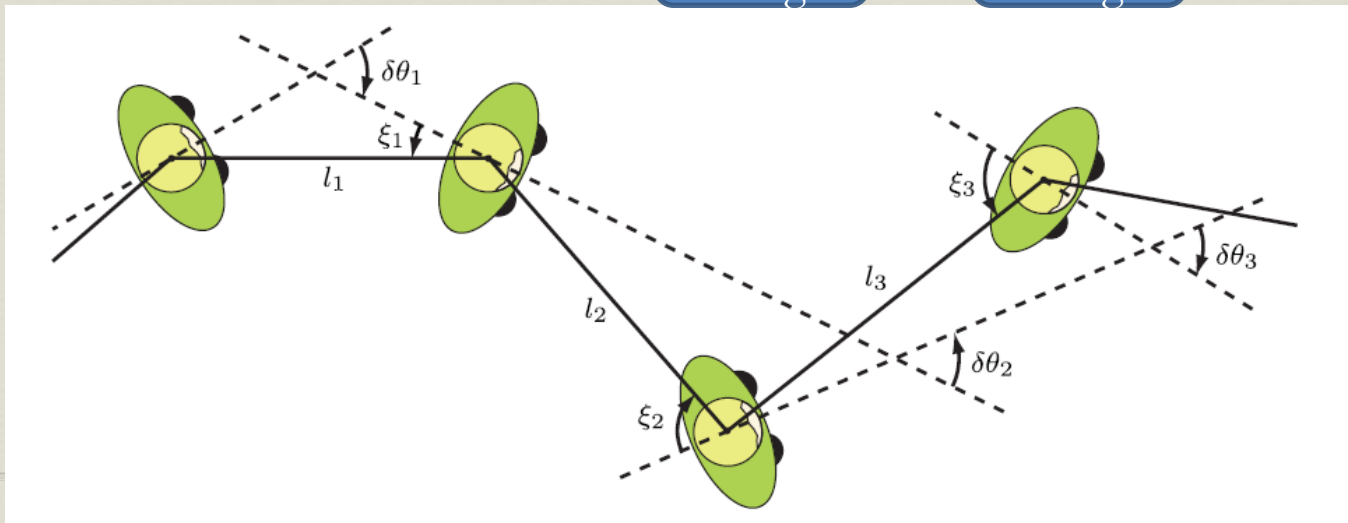
- In practice few systems or applications require updates at such a high frequency.
- Step events need to be detected for collaborating with other filtering algorithms.
- Parameterize a step event by: $e_t = (l_t, \delta z_t, \delta\theta_t, \xi_t)$

Step length

Height change

Heading change

Offset between heading and step direction



Step Segmentation: PDR filter

- Step events are generated at the end of each stance phase.

procedure GENERATESTEP($(\hat{\boldsymbol{C}}_t, \hat{\boldsymbol{s}}_t, \hat{\boldsymbol{v}}_t)$, $(\hat{\boldsymbol{C}}_{t-\delta t}, \hat{\boldsymbol{s}}_{t-\delta t}, \hat{\boldsymbol{v}}_{t-\delta t})$)

// Calculate the change in displacement

$\delta\hat{\boldsymbol{s}} \leftarrow \hat{\boldsymbol{s}}_t - \hat{\boldsymbol{s}}_{t-\delta t}$

// Assume that the user's heading ($\boldsymbol{h}$) is aligned with the x-axis of the IMU

$\boldsymbol{h}^b \leftarrow (1, 0, 0)^T$

// Calculate $\boldsymbol{h}$ in the local frame at the start ($\boldsymbol{h}^g_{t-\delta t}$) and end ($\boldsymbol{h}^g_t$) of the step

$\boldsymbol{h}^g_{t-\delta t} \leftarrow \hat{\boldsymbol{C}}_{t-\delta t}\boldsymbol{h}^b$

$\boldsymbol{h}^g_t \leftarrow \hat{\boldsymbol{C}}_t\boldsymbol{h}^b$

// Calculate the change in the user's heading

$\delta\theta_t \leftarrow \text{atan2}(\boldsymbol{h}^g_t[1], \boldsymbol{h}^g_t[0]) - \text{atan2}(\boldsymbol{h}^g_{t-\delta t}[1], \boldsymbol{h}^g_{t-\delta t}[0])$

// Calculate the step offset

$\xi_t \leftarrow \text{atan2}(\boldsymbol{h}^g_t[1], \boldsymbol{h}^g_t[0]) - \text{atan2}(\delta\hat{\boldsymbol{s}}[1], \delta\hat{\boldsymbol{s}}[0])$

// Calculate the step length and vertical displacement

$l_t \leftarrow \sqrt{\delta\hat{\boldsymbol{s}}[0] * \delta\hat{\boldsymbol{s}}[0] + \delta\hat{\boldsymbol{s}}[1] * \delta\hat{\boldsymbol{s}}[1]}$

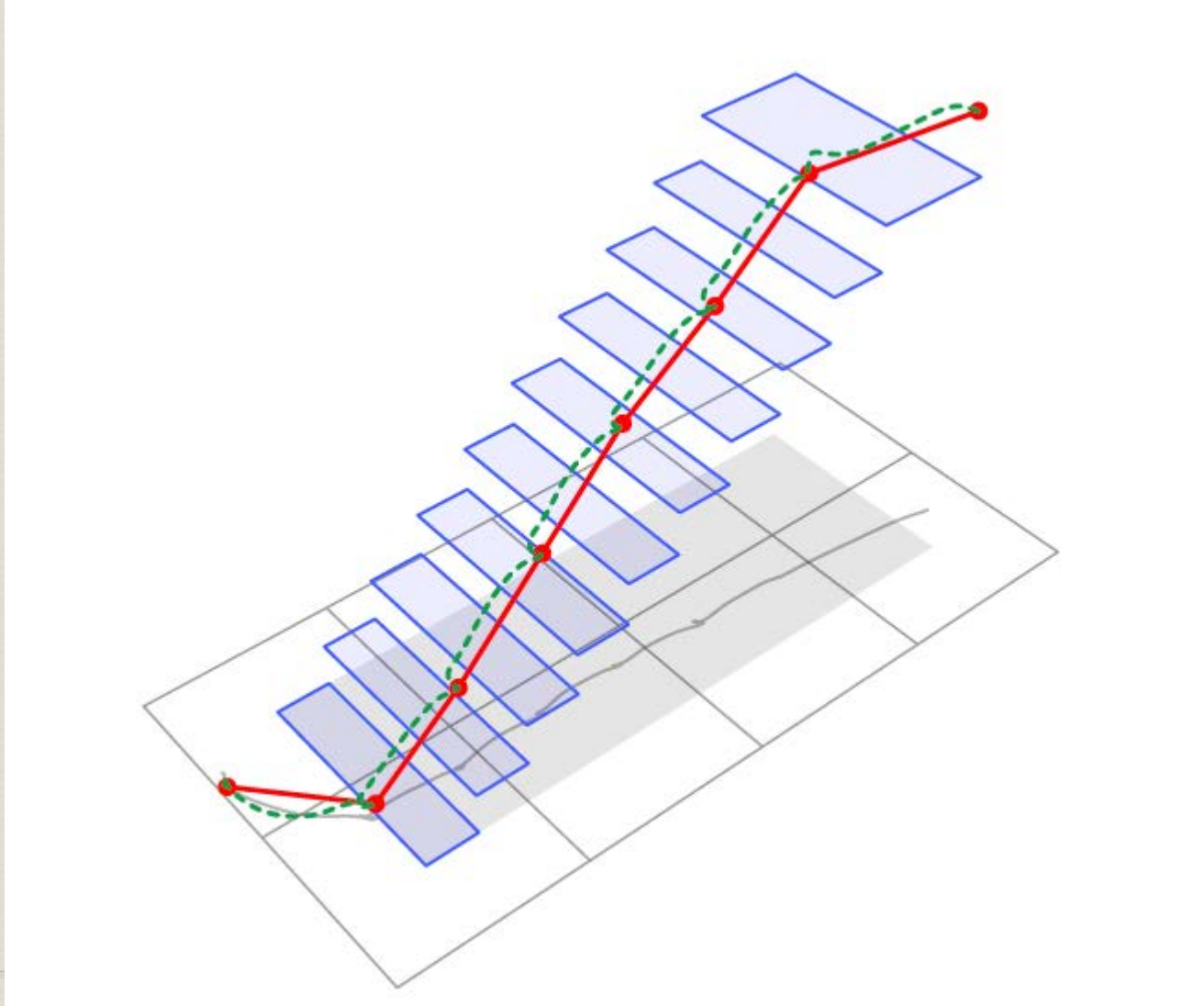
$\delta z_t \leftarrow \delta\hat{\boldsymbol{s}}[2]$

// Return the step event

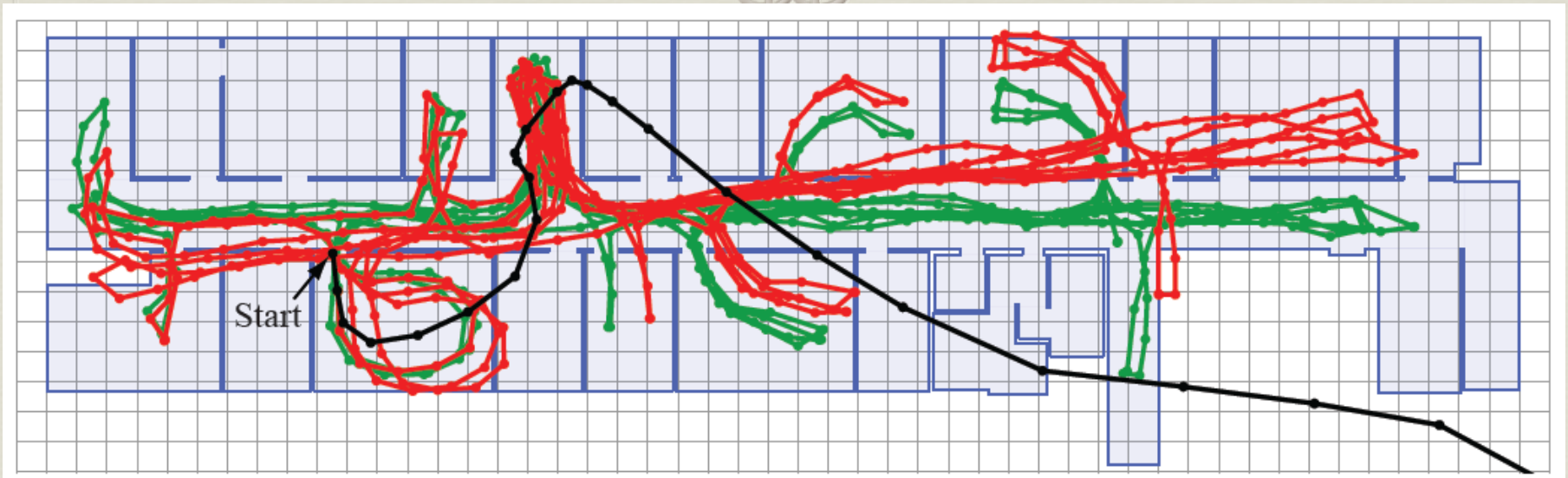
return $(l_t, \delta z_t, \delta\theta_t, \xi_t)$

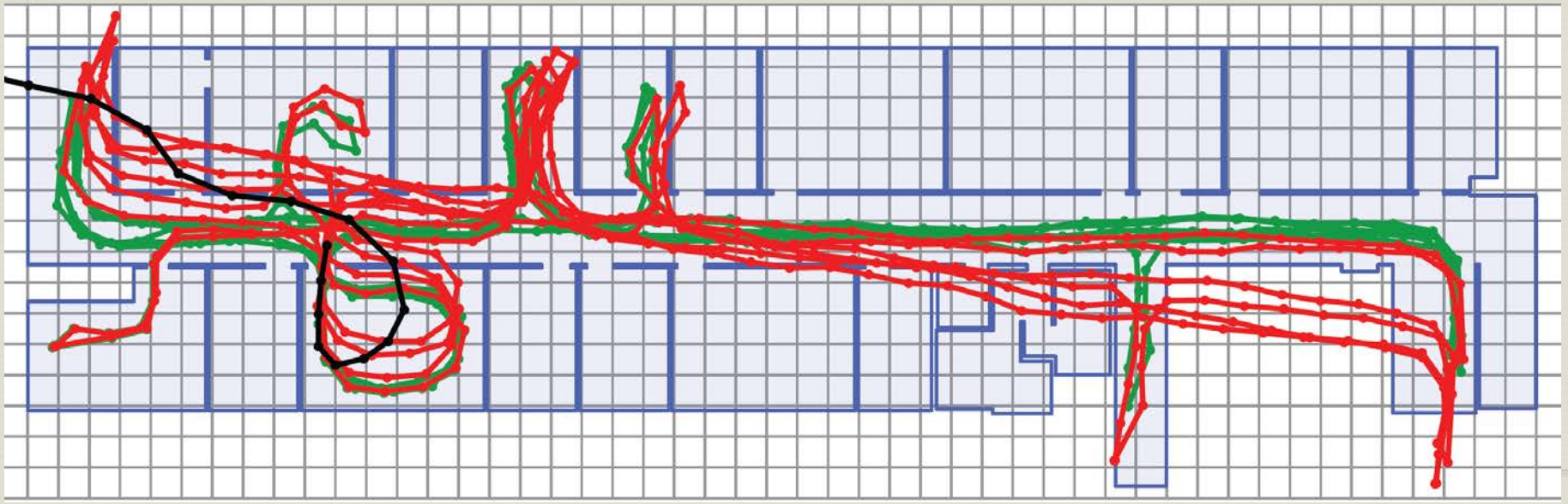
end procedure

INS vs. PDR Filter

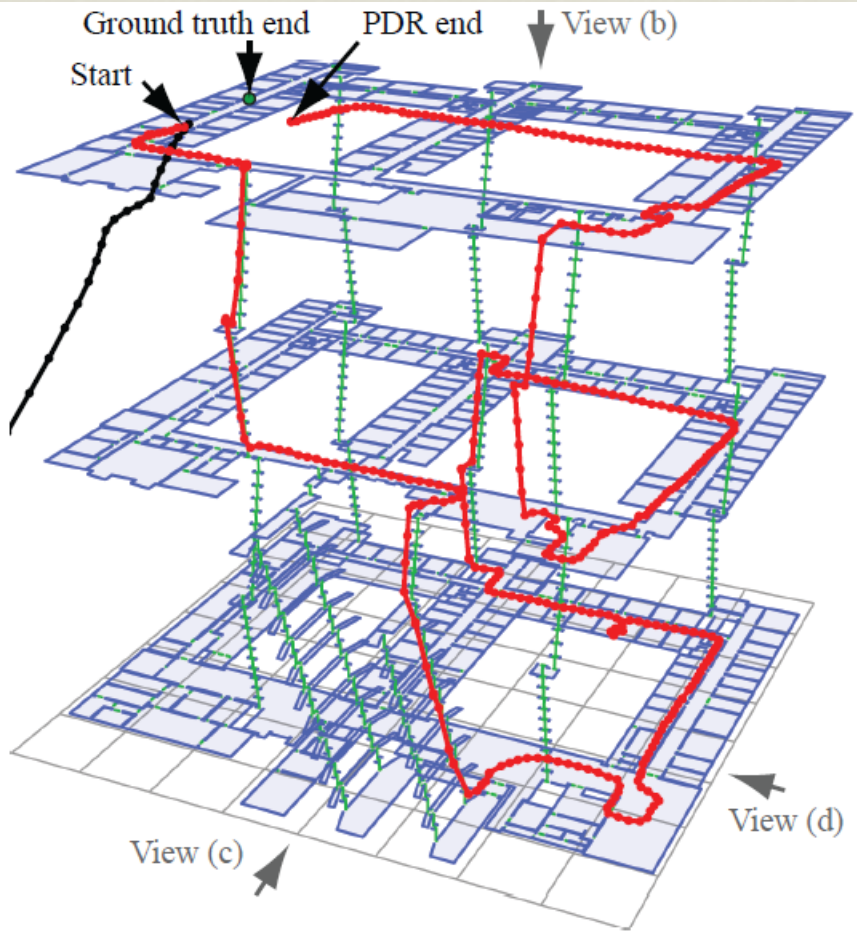


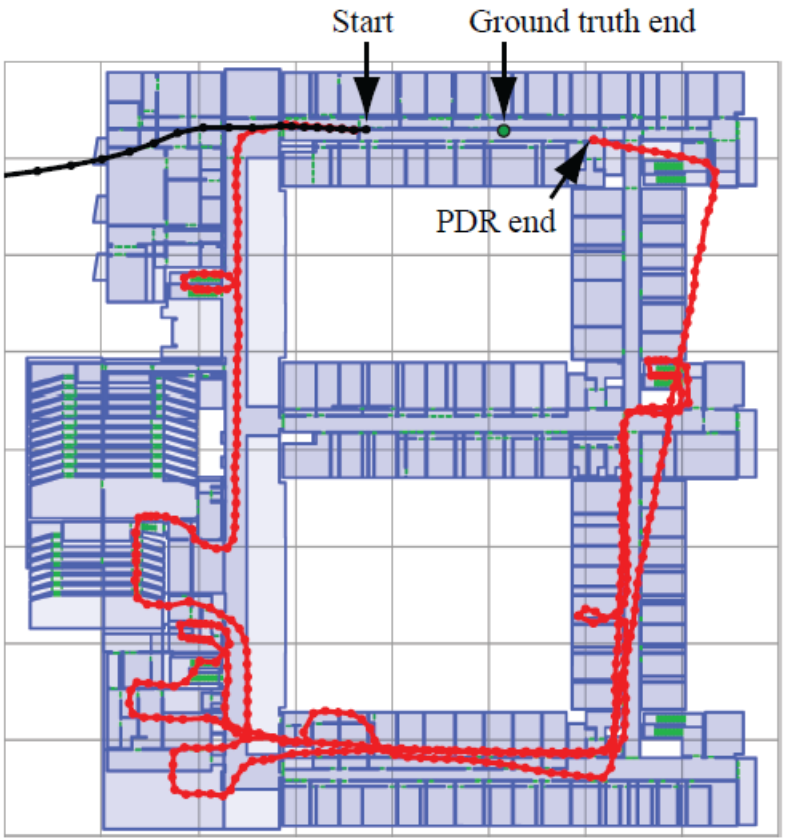
Some Evaluation Results in [2]





2.5D view


Ground truth end
PDR end
View (b)
Start
View (c)
View (d)

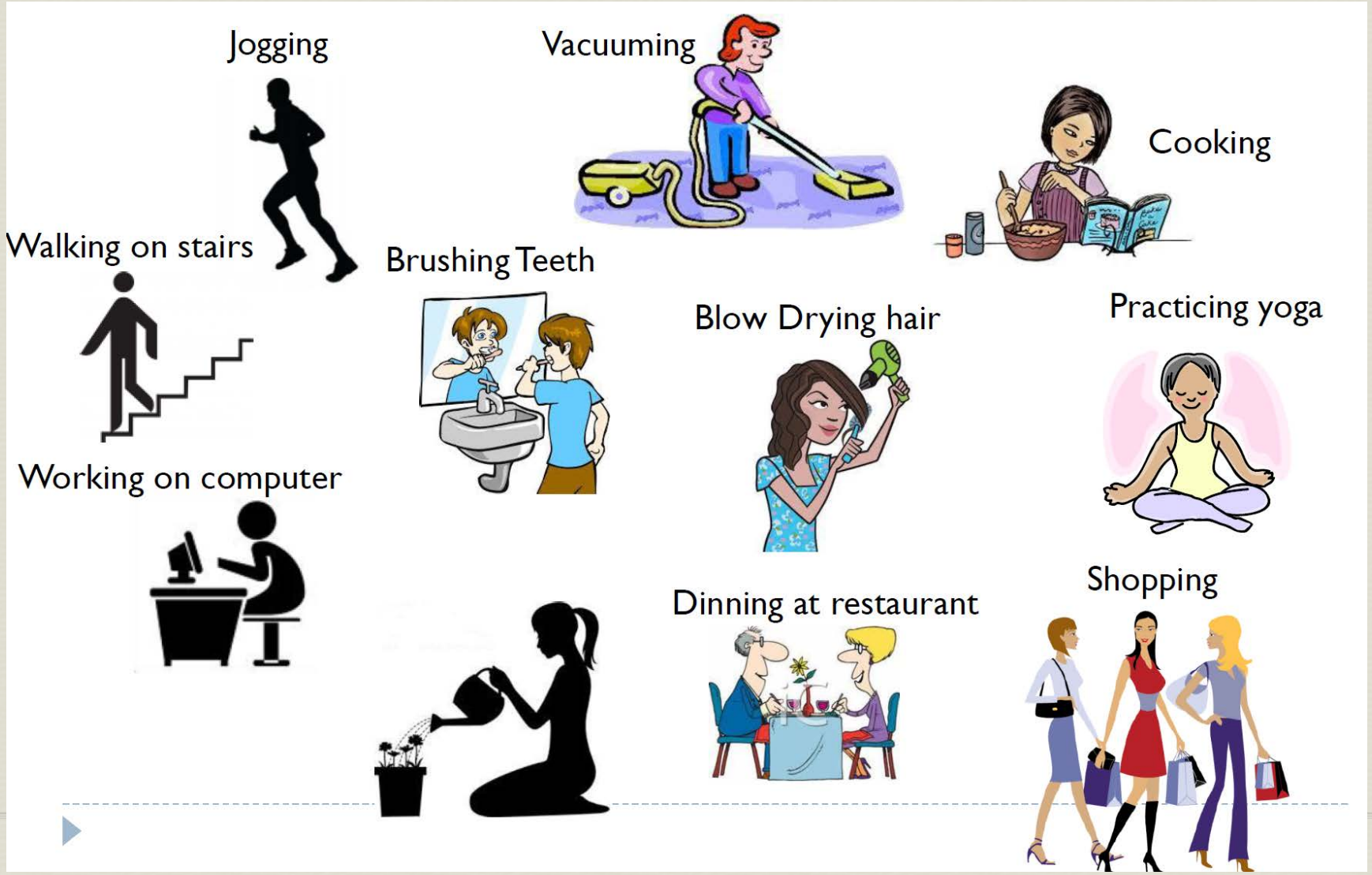

Start
Ground truth end
PDR end

Other Related works

- Using particle filters to correct PDR using map constraints [woodman 2010].
- Using magnetometers to correct PDR [zhou 2014, mobicom]
- Using WiFi radiomap to correct PDR [Miyazaki 2014, Ubicomp]

Other cognitive problems using IMU

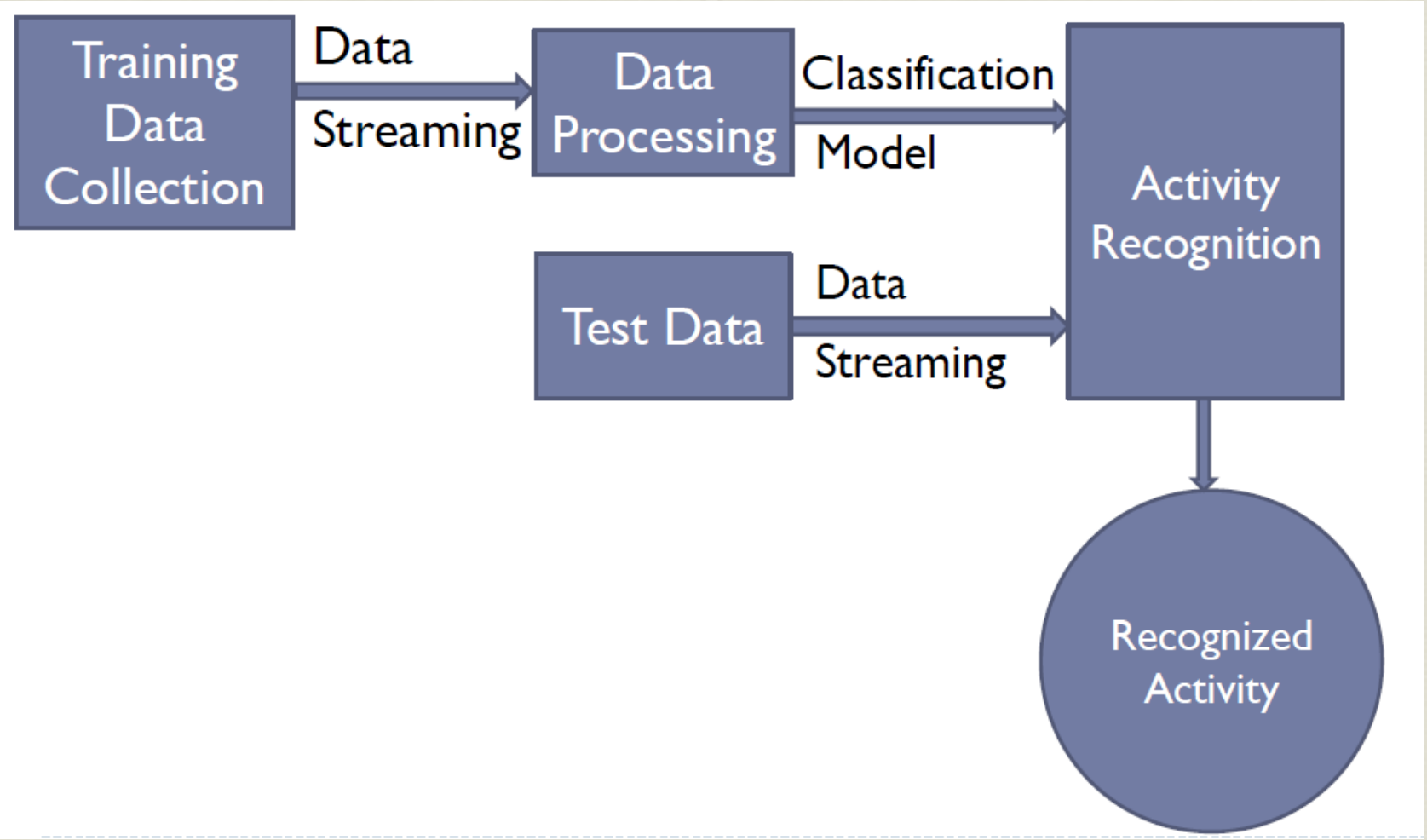
- State recognition



Activities

	Category	Activity Type
Complexity	Simple Activity	Walking, Jogging, Sitting, Standing, Lying, Walking Upstairs, Walking Downstairs, Jumping, Taking escalator up, Taking escalator down, Taking elevator up, Taking elevator down
	Complex Activity	Shopping, Taking buses, Driving a car
Scenario	Living Activity	Brushing Teeth, Vacuuming, Eating, Cooking, Washing hands, Meditation, Clapping hands, Watering plants, Sweeping, Shaving, Blow drying hair, Washing dishes, Ironing, Flushing the toilet, Cleaning
	Working Activity	Working on PC, Talking on the phone, On a break, Meeting, Typing, Writing, Doing a presentation
	Health Activity	Exercising, Fall, Rehabilitation activities, Following routines

Process of Activity Recogonition

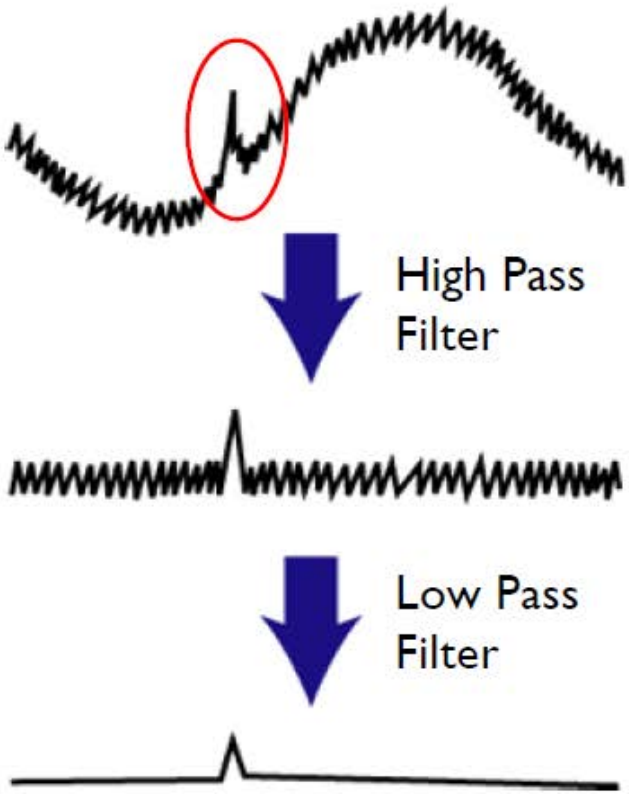


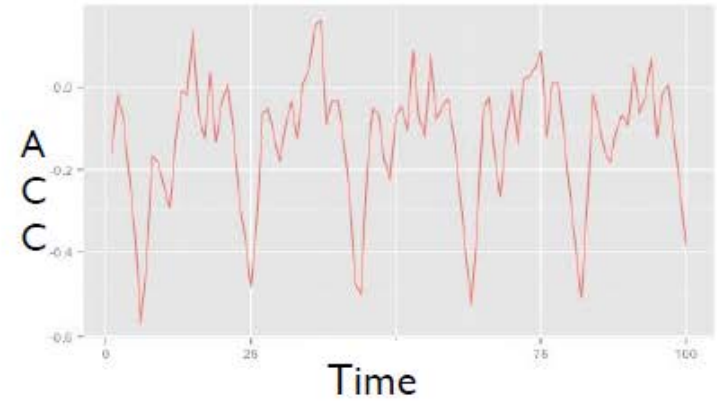
Data collection

- Locations
  - Single location
  - Multiple location

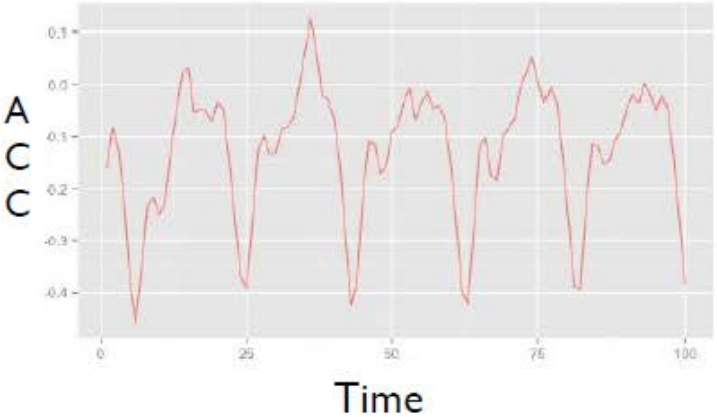
Preprocessing

- De-noising
  - Signal Filter
  - Average





Acceleration Before Smoothing



Acceleration after Smoothing

Time domain Feature Extraction

- Time-Domain Features
  - Mean
  - Max, Min
  - Standard Deviation, Variance
  - Signal-Magnitude Area [8]
    - $\sum_{i=1}^{n} \sqrt{x_i^2 + y_i^2 + z_i^2}$
  - Correlation:
    - $\frac{cov(x,y)}{\sigma_x \sigma_y}$

Frequency domain features

- Frequency Domain Features
  - Energy: $\sum_{i=0}^{n} \sqrt{A\_real_i^2 + A\_image_i^2}$, n is the window's size. [9]
  - Entropy: Normalized information entropy of the discrete FFT component [10]
  - Time Between Peaks[11]
  - Binned Distribution: histogram of FFTs [11]

Classification

- Base-level Classification
- Meta-level Classification

Base level classifier

- **Decision Tree**
- **Decision Table**
- **K-Nearest Neighbor(KNN)**
- **Hidden Markov Model(HMM)**
- **Support Vector Machine(SVM)**
- **Naïve Bayes, Artificial Neural Networks, Gaussian Mixture Model, etc**

Meta-level classifier

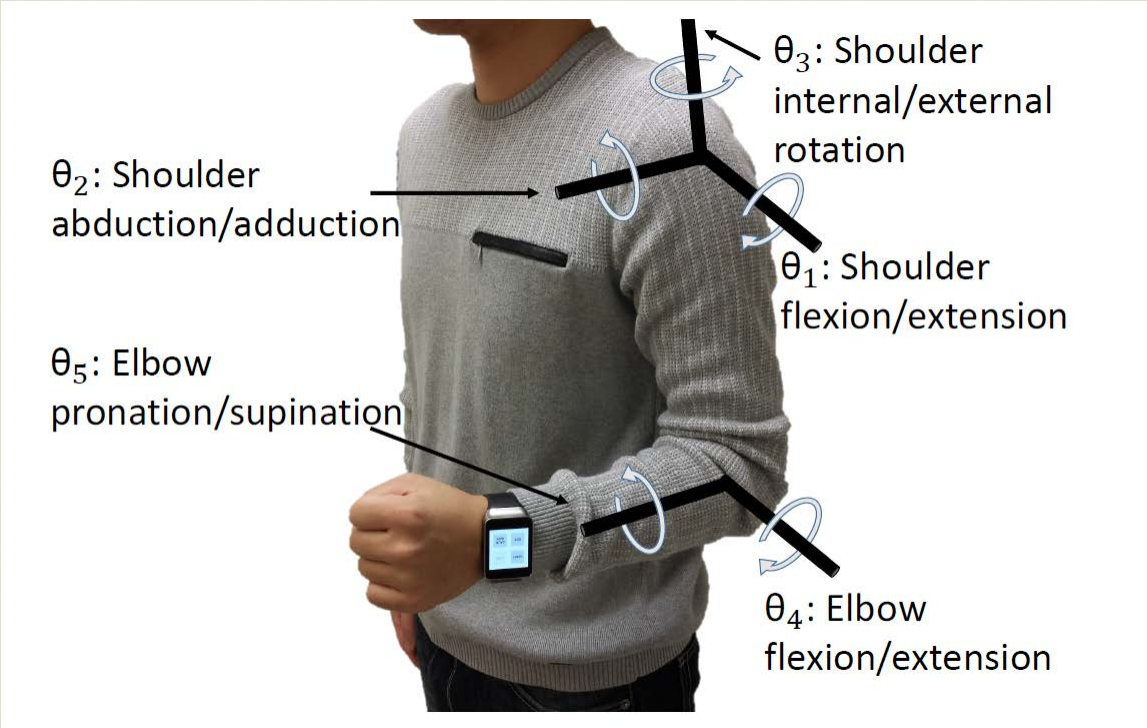
- **Voting**
  - Each base level classifier gives a vote, the class label receiving the most votes is the final decision.
- **Stacking**
  - Learning algorithm to learn how to combine the predictions of the base-level classifiers.
- **Cascading**
  - Iterative process to combine base-level classifiers.
    - Sub-optimal compare to others

Applications

- Calorie consumption estimation
- Fall detection
- Daily life logging
- Sport training
- Gait analysis for healthcare

Cognitive problems using IMU

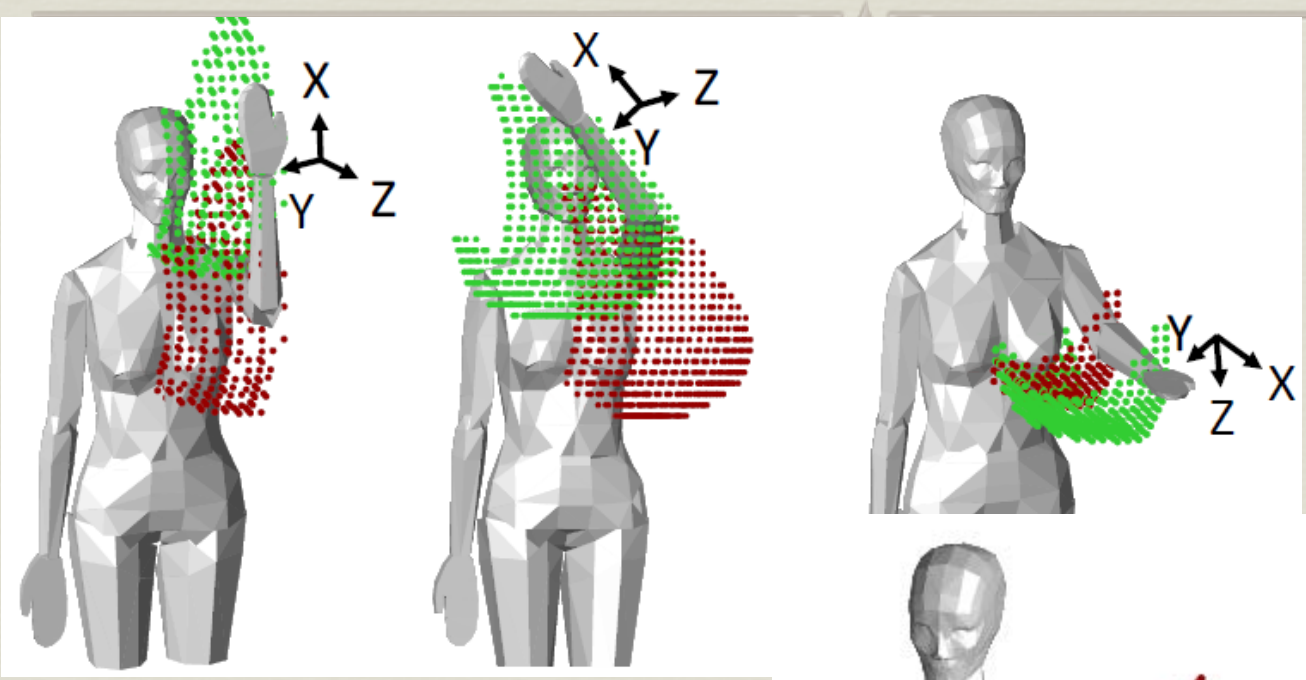
- Arm tracking by smart watch

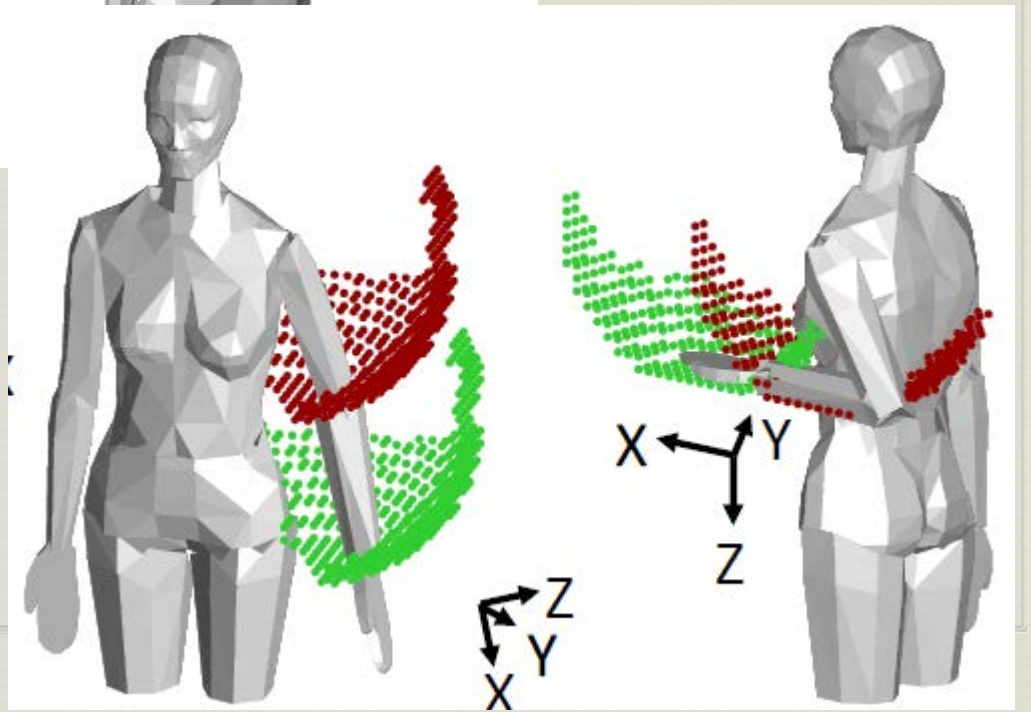


5DOF model of Arm

[3] I am a Smartwatch and I can Track my User's S. Shen, Mobicom 2016

Arm tracking by smart watch





Point cloud model of elbow and whist

Home Work

1. Develop an PDR filter (Step Segmentation) based on the openshoe project.

http://www.openshoe.org/?page_id=362

- Compare the tracking accuracy of Step-based and INS-based
- Test the algorithms using self-collected IMU data traces.