4. The "surrounding infrastructure"

Besides models for dynamics, sensors and world, a successful sensor fusion solution heavily relies on a well functioning "surrounding infrastructure".

This includes for example:

- Time synchronization of the measurements from the different sensors
- Mounting of the sensors and calibration
- Computer vision, radar processing
- Etc...

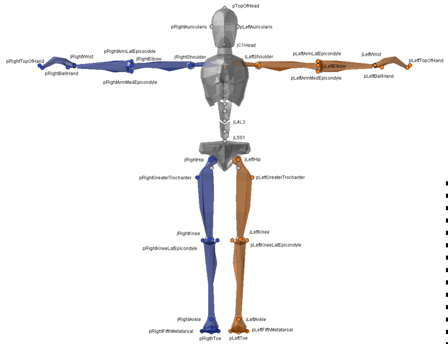
An example:

Relative pose calibration:

Compute the relative translation and rotation of the camera and the inertial sensors that are rigidly connected.

Jeroen D. Hol, Thomas B. Schön and Fredrik Gustafsson. **Modeling and Calibration of Inertial and Vision Sensors**. *International Journal of Robotics Research (IJRR)*, 29(2): 231-244, February 2010.

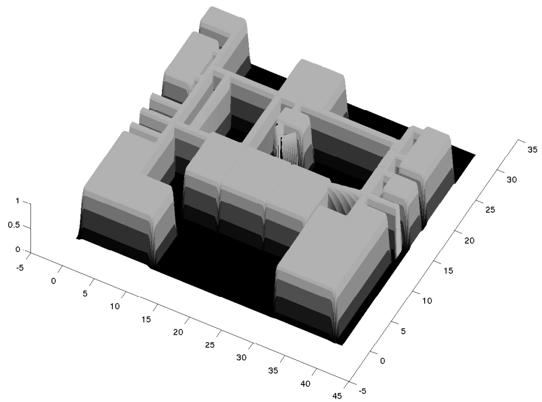
The story I am telling



$$\dot{x} = f(x, u, \theta)$$

1. We are dealing with dynamical systems

This requires a **dynamical model**.



2. The dynamical systems exist in a context.

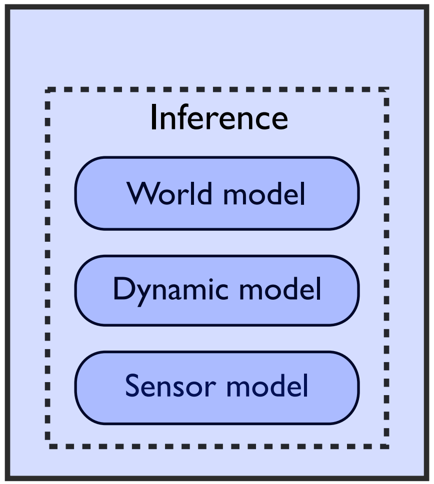
This requires a **world model**.

3. The dynamical systems must be able to perceive their own (and others') motion, as well as the surrounding world.

This requires sensors and **sensor models**.

4. We must be able to transform the measurements from the sensors into knowledge about the dynamical systems and their surrounding world.

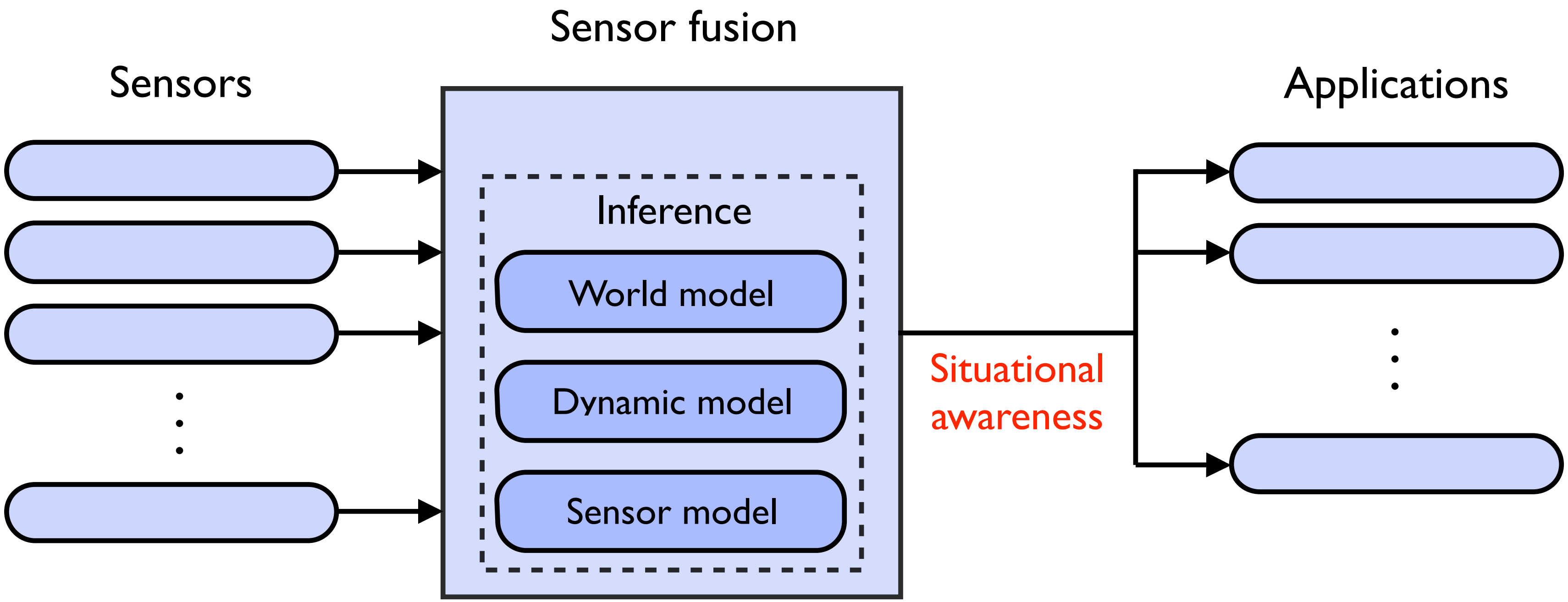
This requires **inference**.



Sensor fusion - definition

Definition (sensor fusion)

Sensor fusion is the process of using information from **several different** sensors to **infer** what is happening (this typically includes finding states of dynamical systems and various static parameters).



Outline

Sensor fusion

1. Probabilistic models of dynamical systems
2. Probabilistic models of sensors and the world
3. Formulate and solve the state inference problem
4. Surrounding infrastructure

A few words about the particle filter

Industrial application examples:

1. Calibration of a camera and an IMU
2. Autonomous landing of a helicopter
3. Helicopter navigation
4. Fighter aircraft navigation
5. Vehicle motion using night vision
6. Indoor motion capture
7. Indoor positioning

Conclusions

State inference - simple special case

Consider the following special case (Linear Gaussian State Space (LGSS) model)

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + v_t, & v_t &\sim \mathcal{N}(0, Q), \\ y_t &= Cx_t + Du_t + e_t, & e_t &\sim \mathcal{N}(0, R). \end{aligned}$$

or, equivalently,

$$\begin{aligned} x_{t+1} \mid x_t &\sim f(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1} \mid Ax_t + Bu_t, Q), \\ y_t \mid x_t &\sim g(y_t \mid x_t) = \mathcal{N}(y_t \mid Cx_t + Du_t, R). \end{aligned}$$

It is now straightforward to show that the solution to the time update and measurement update equations is given by the Kalman filter, resulting in

$$\begin{aligned} p(x_t \mid y_{1:t}) &= \mathcal{N}\left(x_t \mid \widehat{x}_{t|t}, P_{t|t}\right), \\ p(x_{t+1} \mid y_{1:t}) &= \mathcal{N}\left(x_{t+1} \mid \widehat{x}_{t+1|t}, P_{t+1|t}\right). \end{aligned}$$

State inference - interesting case

Obvious question: what do we do in an interesting case, for example when we have a nonlinear model including a world model in the form of a map?

- Need a general representation of the filtering PDF
- Try to solve the equations

$$p(x_t \mid y_{1:t}) = \frac{g(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})},$$

$$p(x_{t+1} \mid y_{1:t}) = \int f(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})\mathrm{d}x_t,$$

as accurately as possible.

State inference - the particle filter (I/II)

The particle filter provides an approximation of the filter PDF

$$p(x_t \mid y_{1:t})$$

when the state evolves according to an SSM

$$\begin{aligned} x_{t+1} \mid x_t &\sim f(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h(y_t \mid x_t, u_t), \\ x_1 &\sim \mu(x_1). \end{aligned}$$

The particle filter maintains an empirical distribution made up N samples (particles) and corresponding weights

$$\widehat{p}(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t)$$

"Think of each particle as one simulation of the system state. Only keep the good ones."

This approximation converge to the true filter PDF,

Xiao-Li Hu, Thomas B. Schön and Lennart Ljung. **A Basic Convergence Result for Particle Filtering**. *IEEE Transactions on Signal Processing*, 56(4):1337-1348, April 2008.

State inference - the particle filter (II/II)

The weights and the particles in

$$\widehat{p}(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t)$$

are updated as new measurements becomes available. This approximation can for example be used to compute an estimate of the mean value,

$$\widehat{x}_{t|t} = \int x_t p(x_t \mid y_{1:t}) \mathrm{d}x_t \approx \int x_t \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t) \mathrm{d}x_t = \sum_{i=1}^{N} w_t^i x_t^i$$

The theory underlying the particle filter has been developed over the past two decades and the theory and its applications are still being developed at a very high speed. For a timely tutorial, see

A. Doucet and A. M. Johansen. **A tutorial on particle filtering and smoothing: fifteen years later**. In *Oxford Handbook of Nonlinear Filtering*, 2011, D. Crisan and B. Rozovsky (eds.). Oxford University Press.

or my new course on computational inference in dynamical systems

user.it.uu.se/~thosc112/CIDS.html

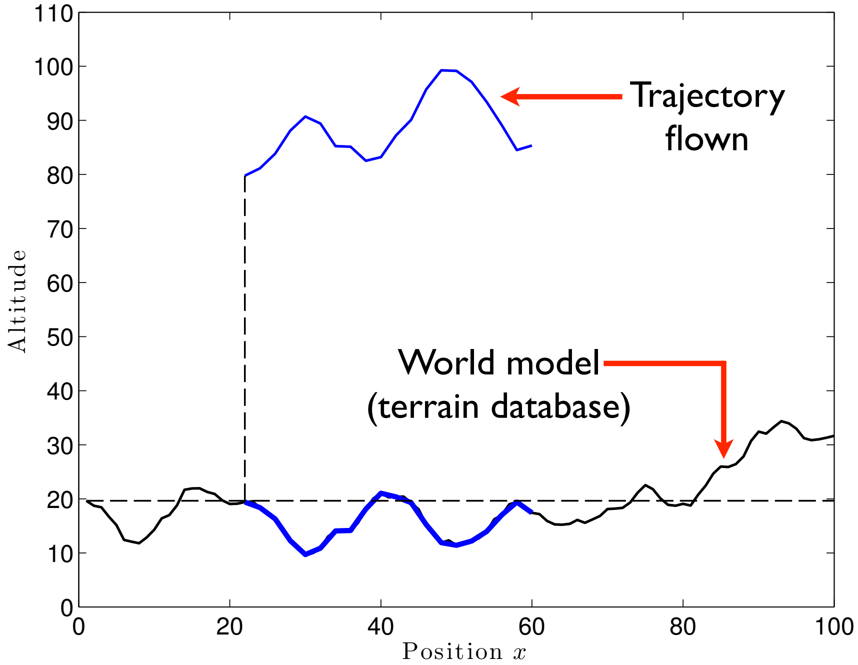
Using world models in solving state inference problems

Consider a 1D localization example.

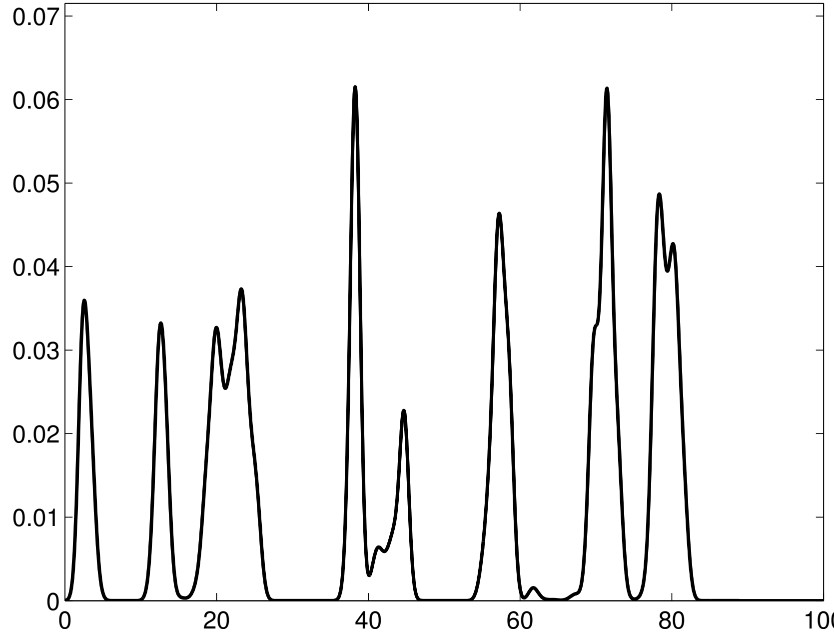
position ↓ velocity (measured input) ↓

$$x_{t+1} = x_t + u_t + v_t,$$
$$y_t = h(x_t) + e_t.$$

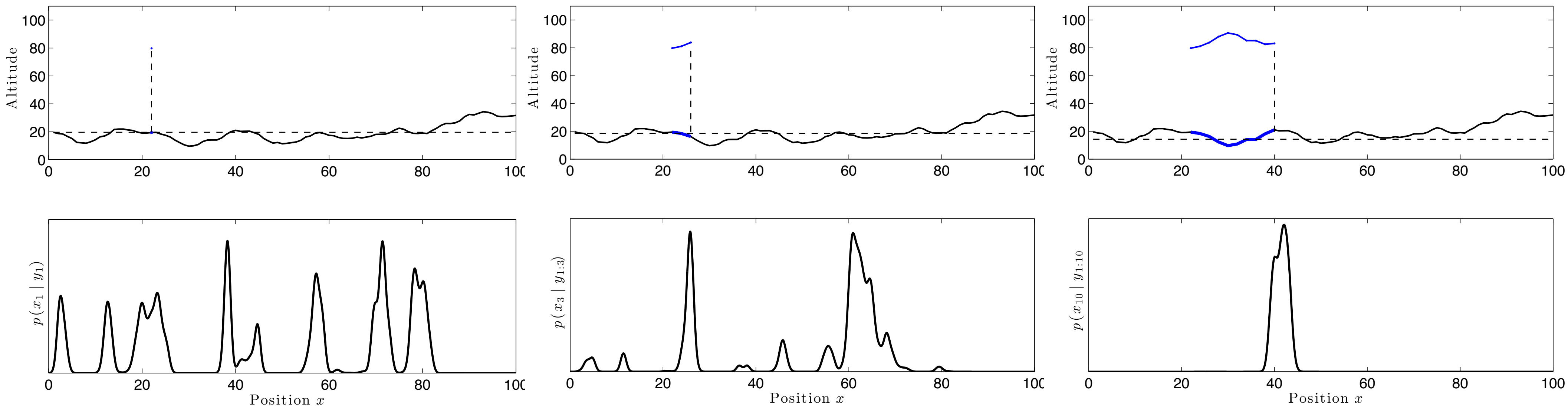
↑ measurement (altitude) ↑ world model (terrain database)



Filter PDF after 1 measurement $p(x_1 \mid y_1)$ →



Using world models in solving state inference problems



Filter PDF after 1 measurement

$p(x_1 \mid y_1)$

Filter PDF after 3 measurements

$p(x_3 \mid y_{1:3})$

Filter PDF after 10 measurements

$p(x_{10} \mid y_{1:10})$

Using world models in solving state inference problems

The simple 1D localization example is an illustration of a problem involving a multimodal filter PDF

- Straightforward to represent and work with using a PF
- Horrible to work with using e.g. an extended Kalman filter

The example also highlights the **key capabilities** of the PF:

1. To automatically handle an unknown and dynamically changing number of hypotheses.

2. Work with nonlinear/non-Gaussian models

We have implemented a similar localization solution for this aircraft (Gripen).

Industrial partner: Saab

Outline

Sensor fusion

1. Probabilistic models of dynamical systems
2. Probabilistic models of sensors and the world
3. Formulate and solve the state inference problem
4. Surrounding infrastructure

A few words about the particle filter

Industrial application examples:

1. Calibration of a camera and an IMU
2. Autonomous landing of a helicopter
3. Helicopter navigation
4. Fighter aircraft navigation
5. Vehicle motion using night vision
6. Indoor motion capture
7. Indoor positioning

Conclusions

1. Calibration of a camera and an IMU

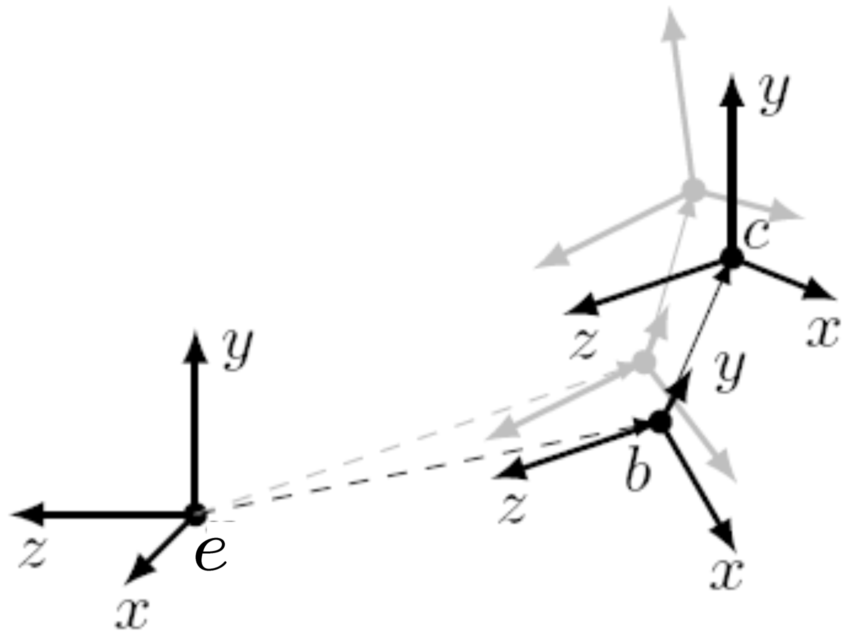
Aim: Compute high quality estimates of the relative position and orientation of a camera and an inertial measurement unit (IMU) that are rigidly mounted.

Industrial partner: Xsens

The resulting algorithm does not require any additional hardware, except a piece of paper with a checkerboard pattern.

Coordinate frames

Earth (e): Fixed frame
Body (b): The coordinate frame where the inertial measurements are obtained.
Camera (c): Attached to the camera.



1. Calibration of a camera and an IMU

Sensor unit developed within the project

The sensor unit consists of:

- IMU
  - Gyroscope (3-D)
  - Accelerometer (3-D)
- Camera

Vision (and existing 3D model)

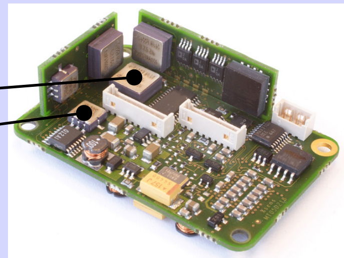
Benefits

- Absolute pose
- Drift-free

Drawbacks

- Only works for slow motions
- Problems with occlusion, etc.
- Requires many correspondences

Inertial sensors



Benefits

- Handles unconstrained motions
- Always works

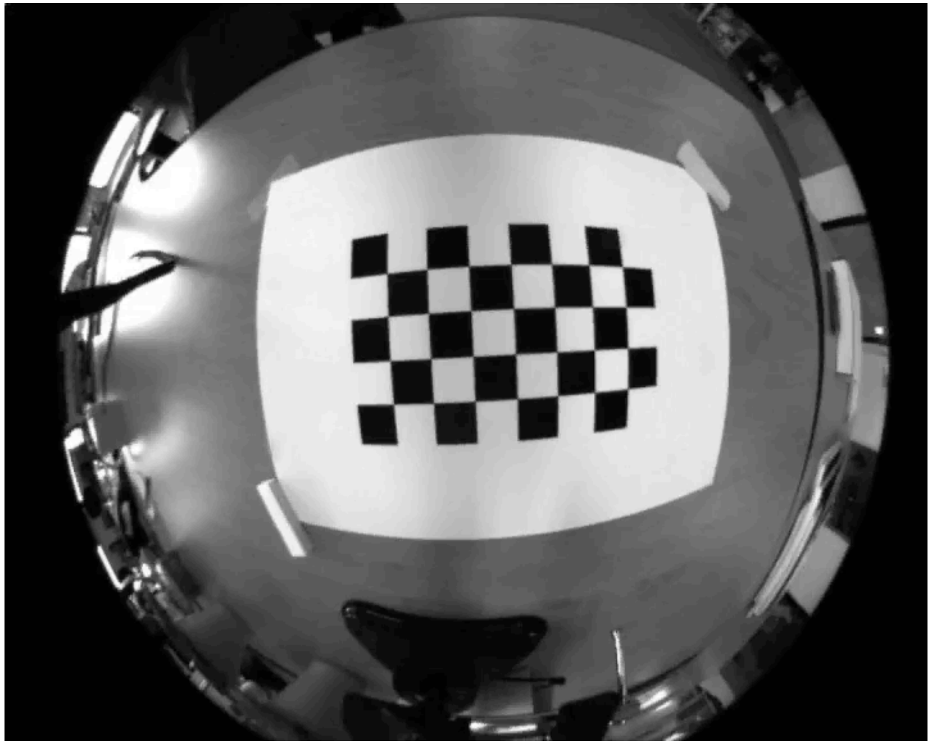
Drawbacks

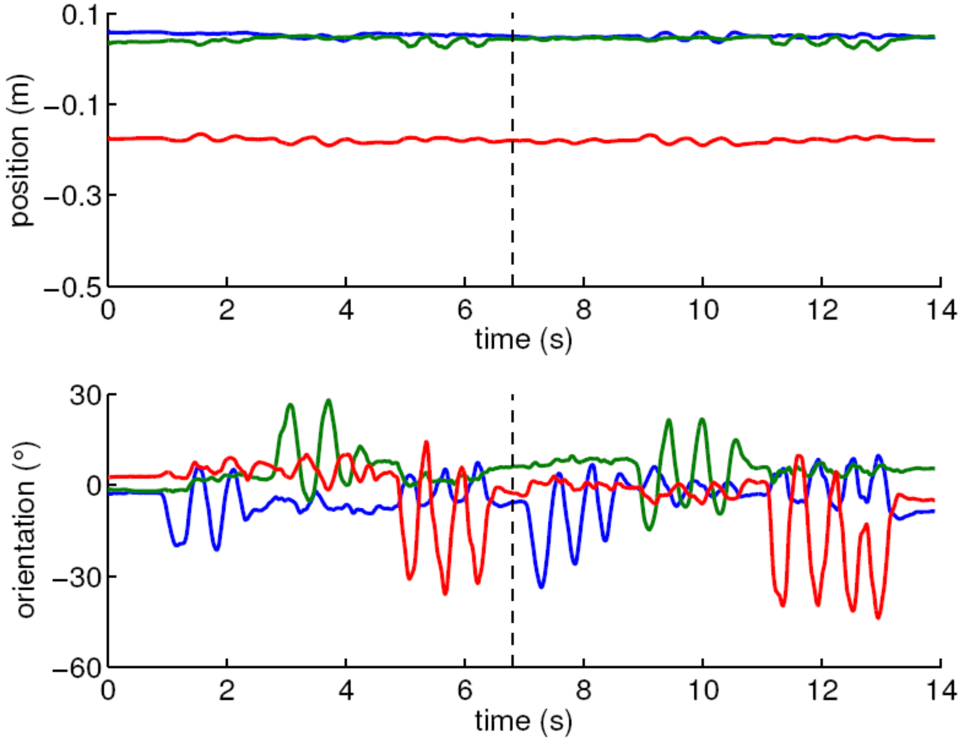
- Big drift
- Only relative measurements

1. Calibration of a camera and an IMU

What is the first step of solving a system identification problem?

$$Z = \{u_1, \dots, u_M, y_1, \dots, y_N\}$$





The camera information is modeled as output $\{y_1, \dots, y_N\}$

The inertial information is modeled as input $\{u_1, \dots, u_M\}$

1. Calibration of a camera and an IMU

We formulate it as a standard gray-box problem

Derive a good predictor $\hat{y}_{t|t-1}(\theta, Z)$

Pose and solve an appropriate optimization problem (NLS)

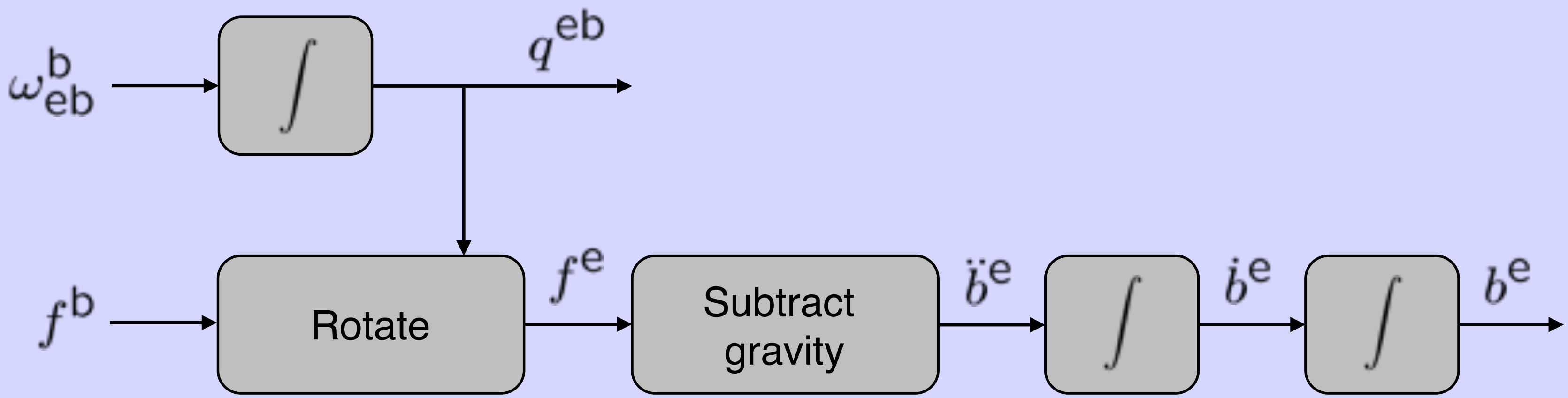
$$\hat{\theta} = \arg\min_{\theta} V_N(\theta, Z)$$

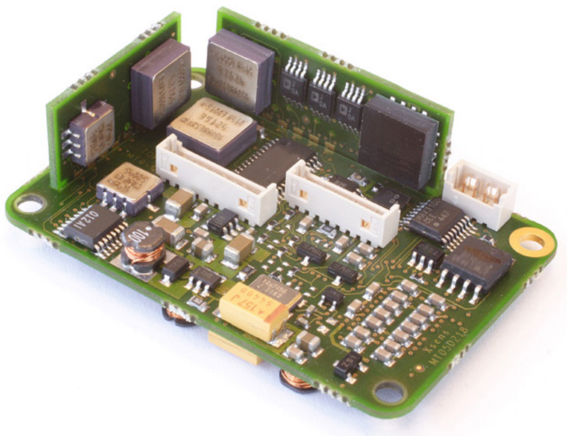
where,

$$V_N(\theta, Z) = \frac{1}{N} \sum_{t=1}^{N} \|y_t - \hat{y}_{t|t-1}(\theta)\|^2_{\Lambda_t}$$

This is a standard gray-box system identification problem!

Inertial navigation:





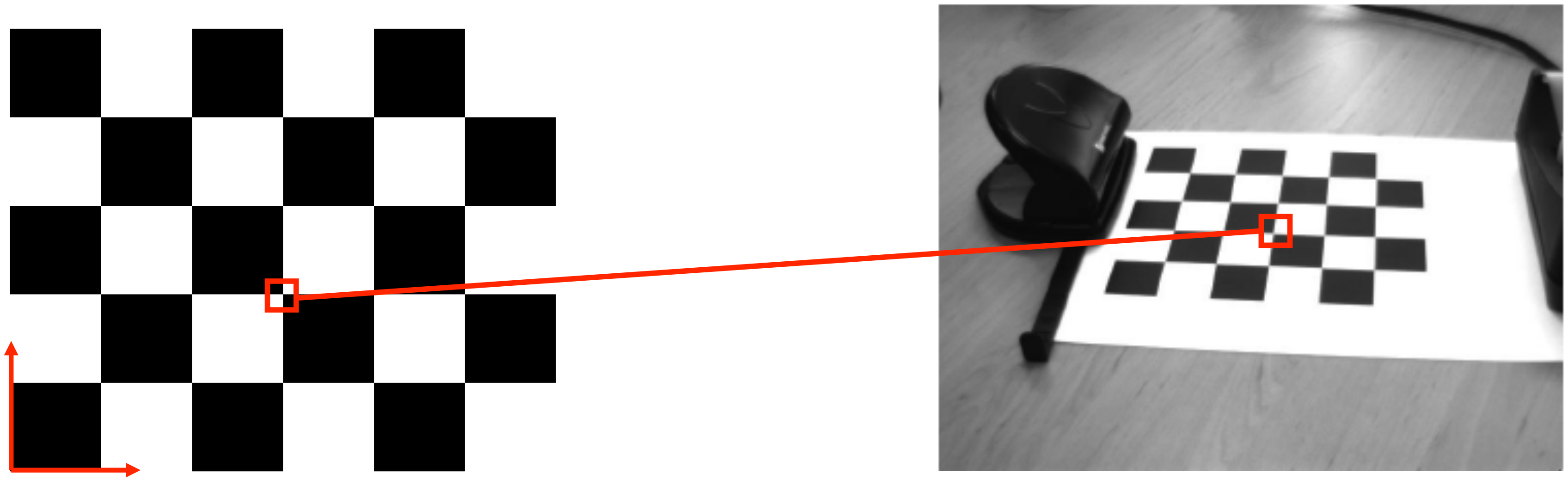
The gyroscopes measures the angular velocities

$$y_\omega = \omega^{\text{b}}_{\text{eb}} + \delta^{\text{b}}_\omega + e^{\text{b}}_\omega$$

The accelerometers measures the specific force

$$y_a = \underbrace{R^{\text{be}}(\ddot{b}^{\text{e}} - g^{\text{e}})}_{f^{\text{b}}} + \delta^{\text{b}}_a + e^{\text{b}}_a$$

1. Calibration of a camera and an IMU



3D model of the scene   Camera image of the scene

The vision measurements consists in 2D-3D correspondences between the 2D camera image and the 3D model of the scene.

2D position in the normalized image frame

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} X \\ Y \end{pmatrix}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R^{cb}(R_t^{be}(p_{t,k}^e - b_t^e) - c^b)$$

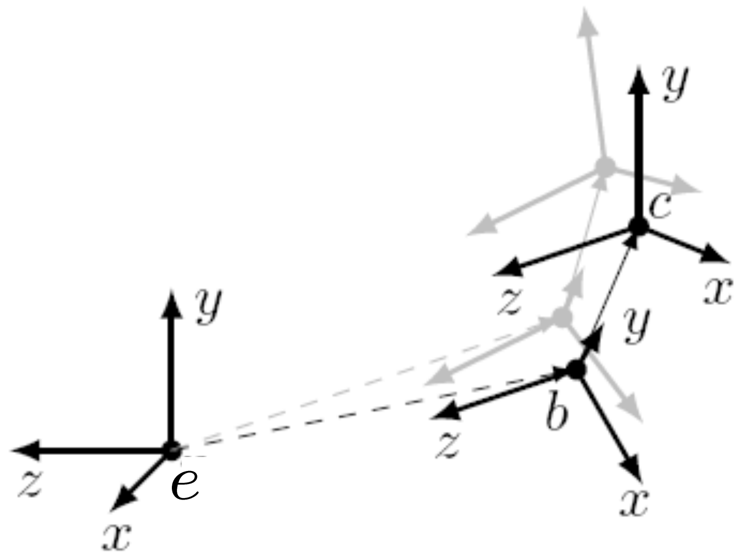
Corresponding 3D position in the camera frame

Coordinate frames:

Earth (e): Fixed frame

Body (b): The coordinate frame where the inertial measurements are obtained

Camera (c): Attached to the camera



Frames b and c are rigidly connected

State vector:

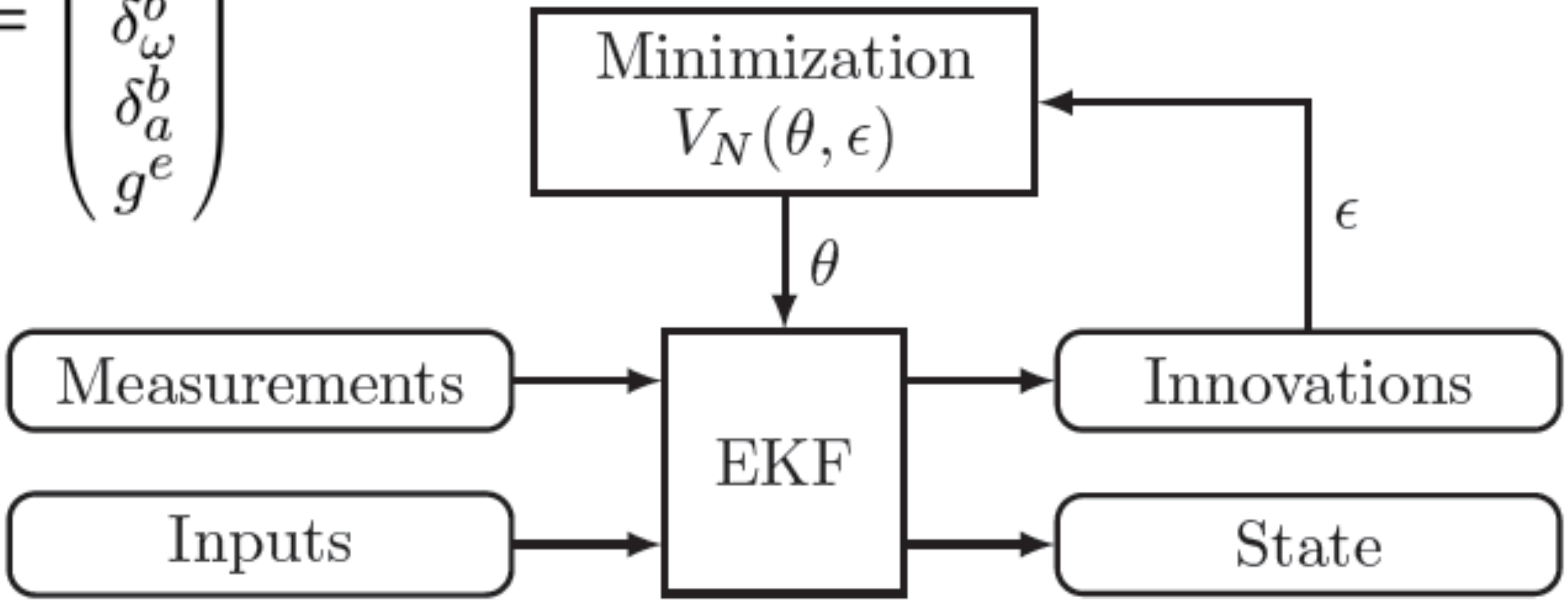
$$x = \begin{pmatrix} b^{\mathrm{e}} \\ \dot{b}^{\mathrm{e}} \\ q^{\mathrm{be}} \end{pmatrix}$$

- b^{e}: Position of the body (b) frame, resolved in the earth (e) frame
- $\dot{b}^{\mathrm{e}}$: Velocity of the body (b) frame, resolved in the earth (e) frame
- q^{be}: Unit quaternion describing the orientation (from e to b)

1. Calibration of a camera and an IMU

Calibration algorithm (standard gray-box identification)

$$\theta = \begin{pmatrix} \varphi^{cb} \\ c^b \\ \delta_\omega^b \\ \delta_a^b \\ g^e \end{pmatrix}$$

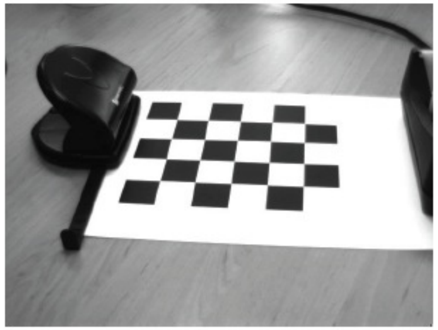


$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \sum_{t=1}^{N} (y_t - \hat{y}_{t|t-1}(\theta))^T S_t(\theta)^{-1} (y_t - \hat{y}_{t|t-1}(\theta))$$

1. Calibration of a camera and an IMU

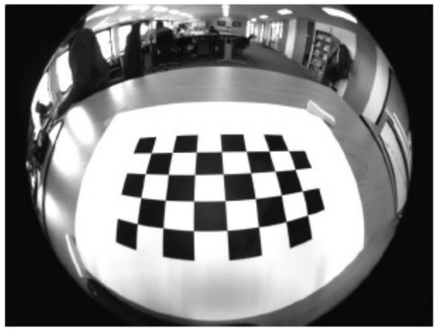
1. Place a camera calibration pattern on a horizontal, level surface.
2. Acquire inertial measurements and images.
   - Rotate around all 3 axes, with sufficiently exciting angular velocities.
   - Always keep the calibration pattern in view.
3. Obtain the point correspondences of the calibration pattern for all images.
4. Compute an estimate $\widehat{\theta}$ by minimizing $V_N(\theta, Z)$

(a)



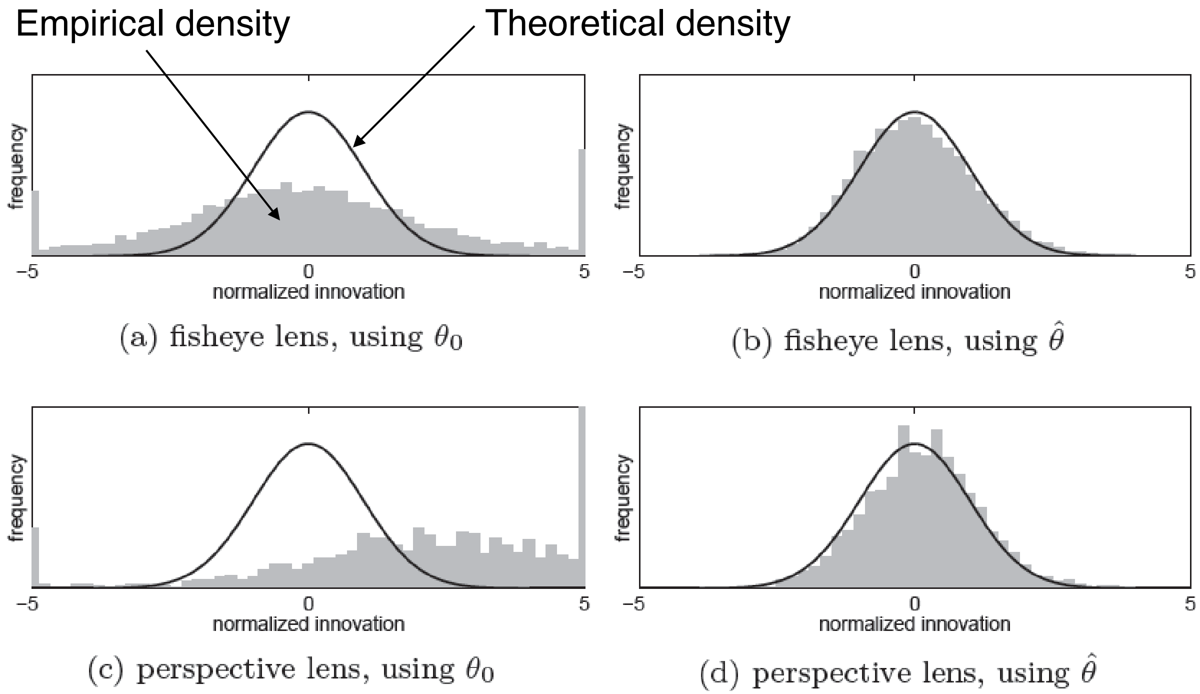
(b)

(c)



(d)

1. Calibration of a camera and an IMU



Histograms of the normalized innovations using validation data.

Full details are available here:

Jeroen D. Hol, Thomas B. Schön and Fredrik Gustafsson. **Modeling and Calibration of Inertial and Vision Sensors**. *International Journal of Robotics Research (IJRR)*, 29(2):231-244, February 2010.

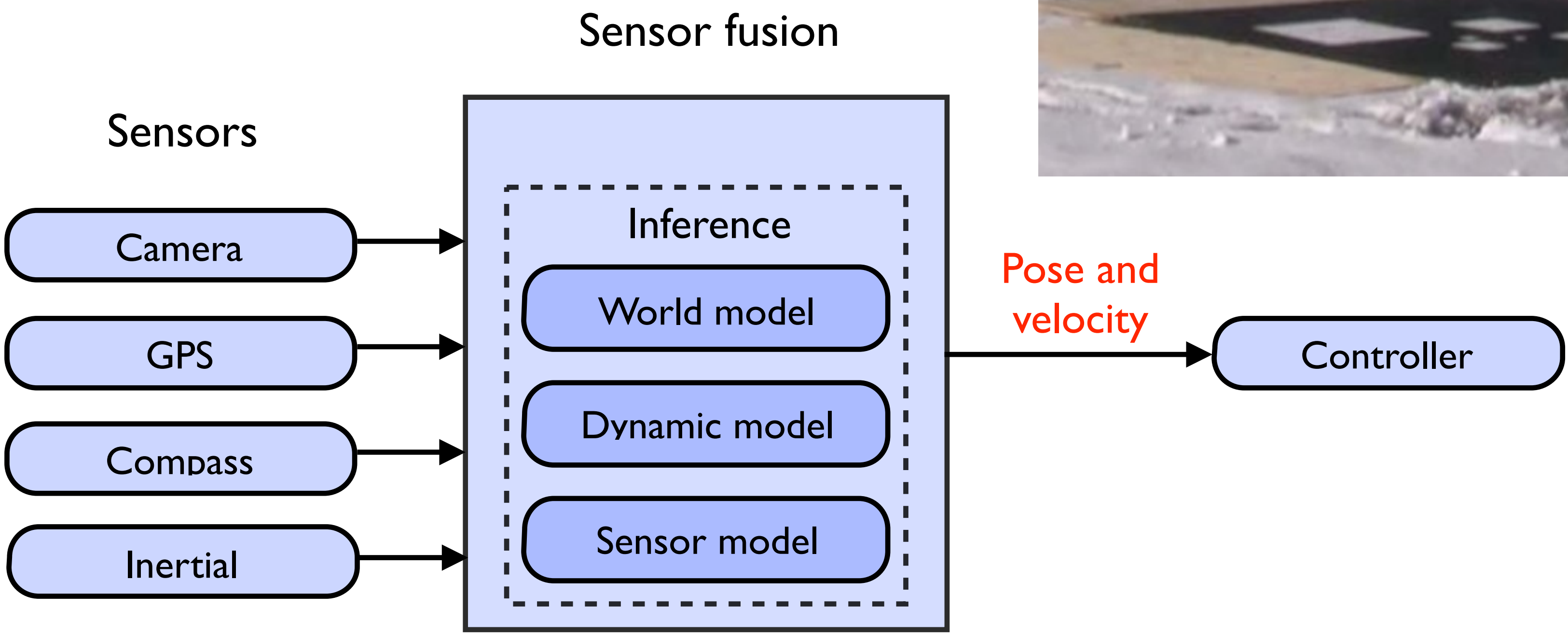
Recently we also solved the problem of calibrating magnetometers and inertial sensors, see

Manon Kok and Thomas B. Schön. **Maximum likelihood calibration of a magnetometer using inertial sensors**. *In Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC)*, Cape Town, South Africa, August 2014.

2. Autonomous helicopter landing (I/III)

Aim: Land a helicopter autonomously using information from a camera, GPS, compass and inertial sensors.

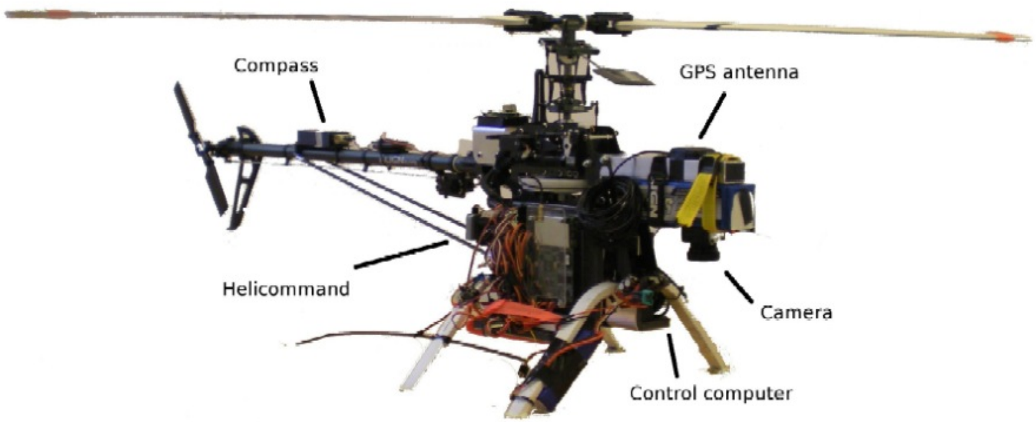
Industrial partner: Cybaero



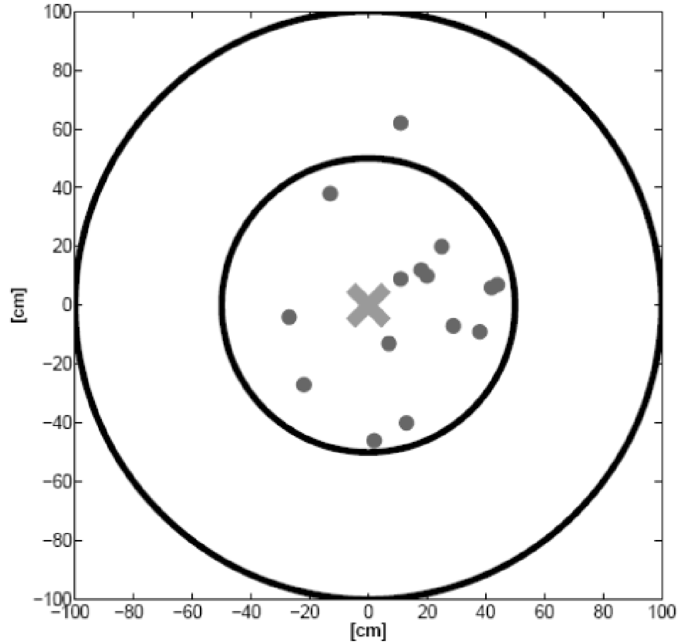
2. Autonomous helicopter landing (II/III)

Experimental helicopter

- Weight: 5kg
- Electric motor



Results from 15 landings

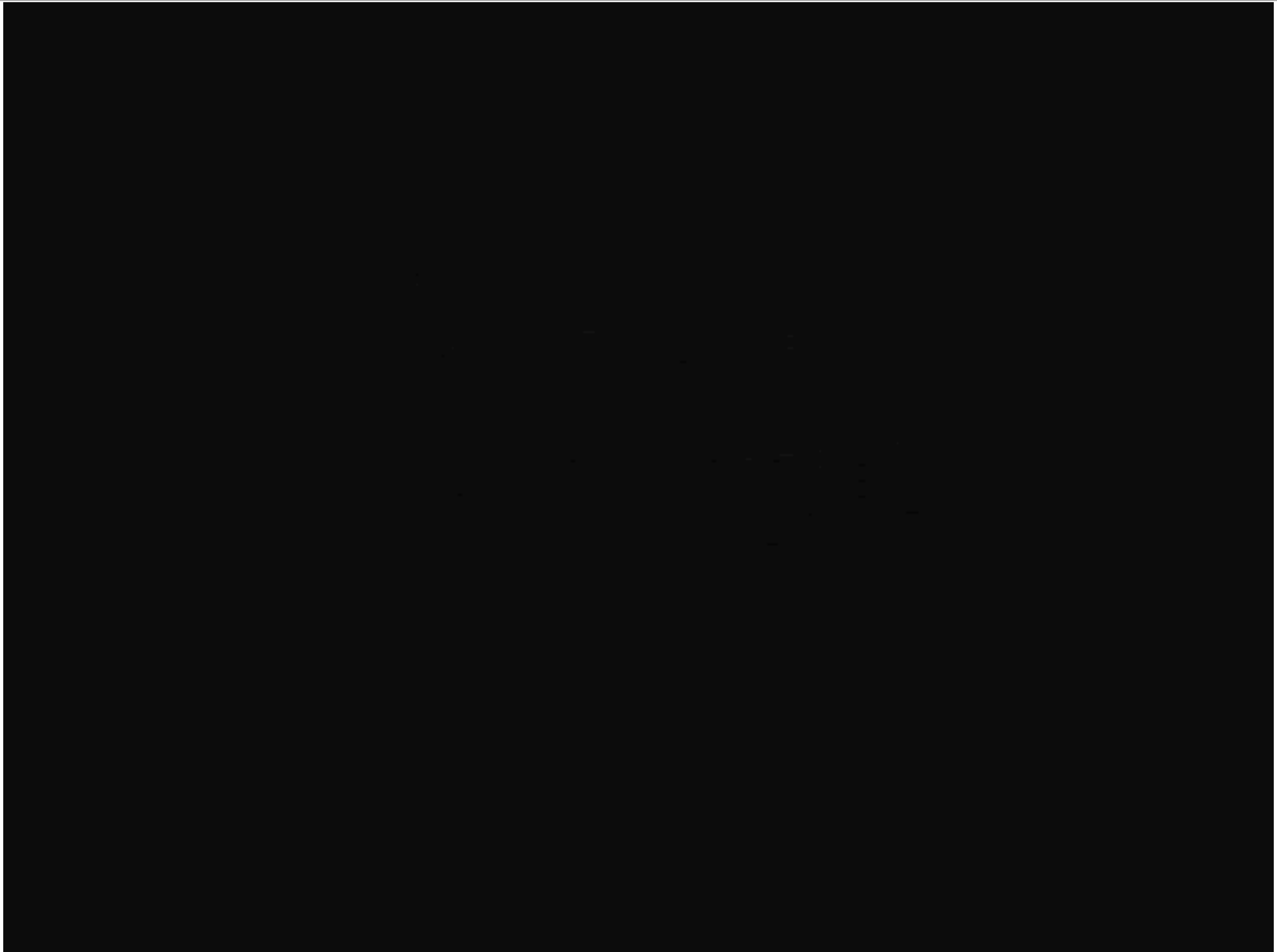


The two circles mark 0.5m and 1m landing error, respectively.

Dots = achieved landings
Cross = perfect landing

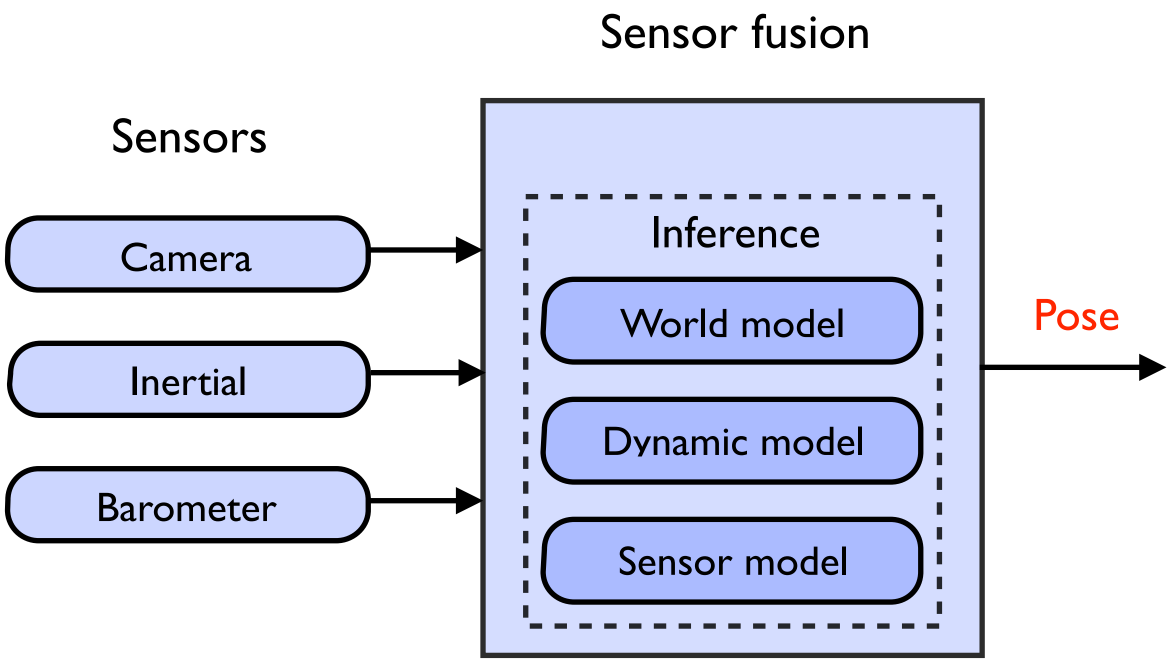
Joel Hermansson, Andreas Gising, Martin Skoglund and Thomas B. Schön. **Autonomous Landing of an Unmanned Aerial Vehicle**. *Reglermöte (Swedish Control Conference)*, Lund, Sweden, June 2010.

2. Autonomous helicopter landing (III/III)



3. Helicopter pose estimation using a map (I/III)

Aim: Compute the position and orientation of a helicopter by exploiting the information present in Google maps images of the operational area.



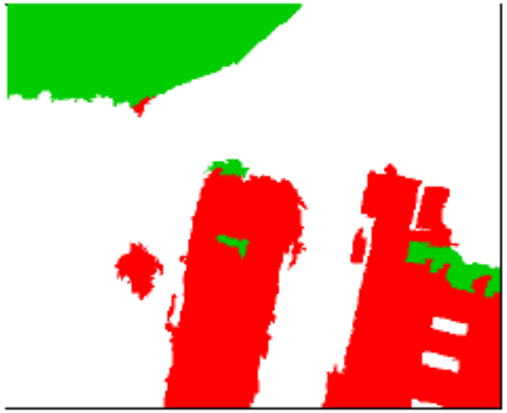
3. Helicopter pose estimation using a map (II/III)

Map over the operational environment obtained from Google Earth.

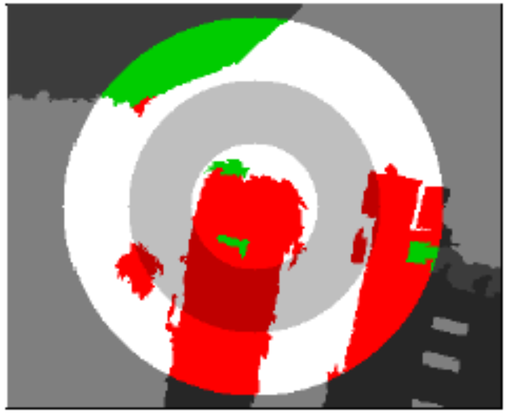
Manually classified map with grass, asphalt and houses as pre-specified classes.

Image from on-board camera

Extracted superpixels

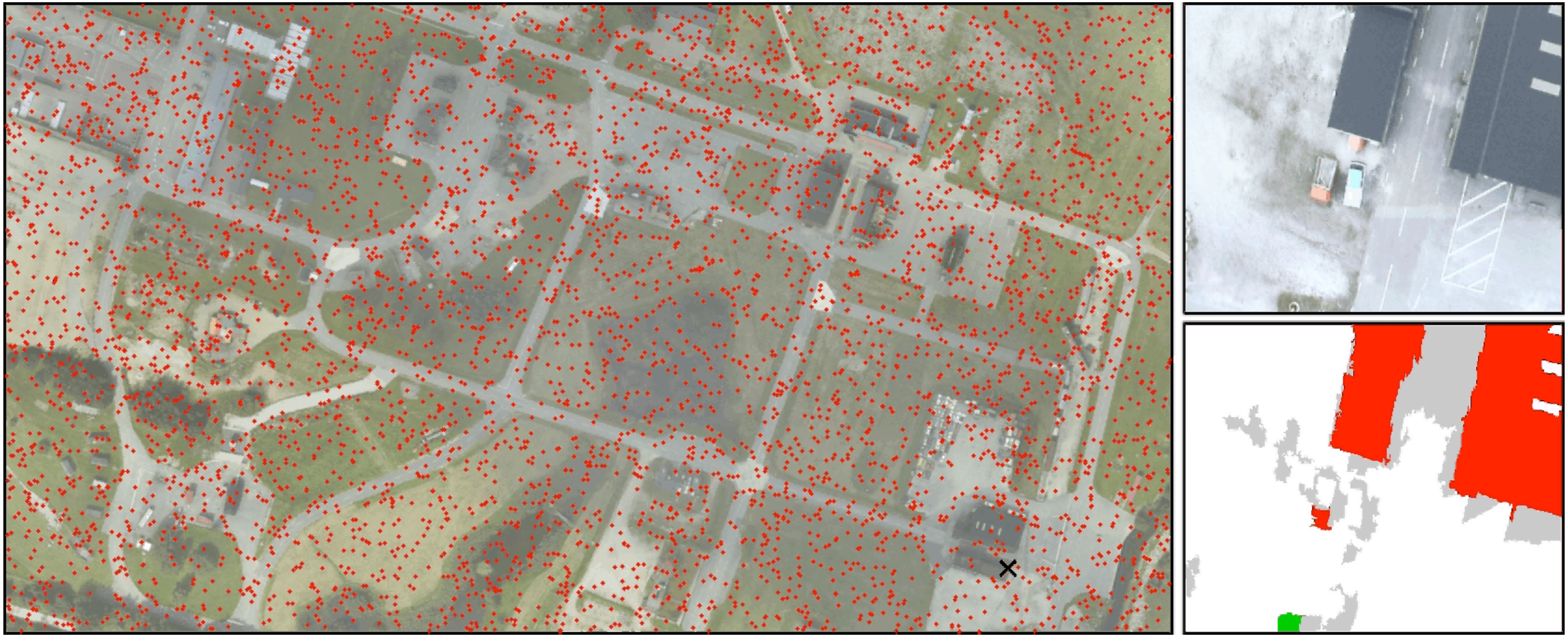


Superpixels classified as grass, asphalt or house



Three circular regions used for computing class histograms

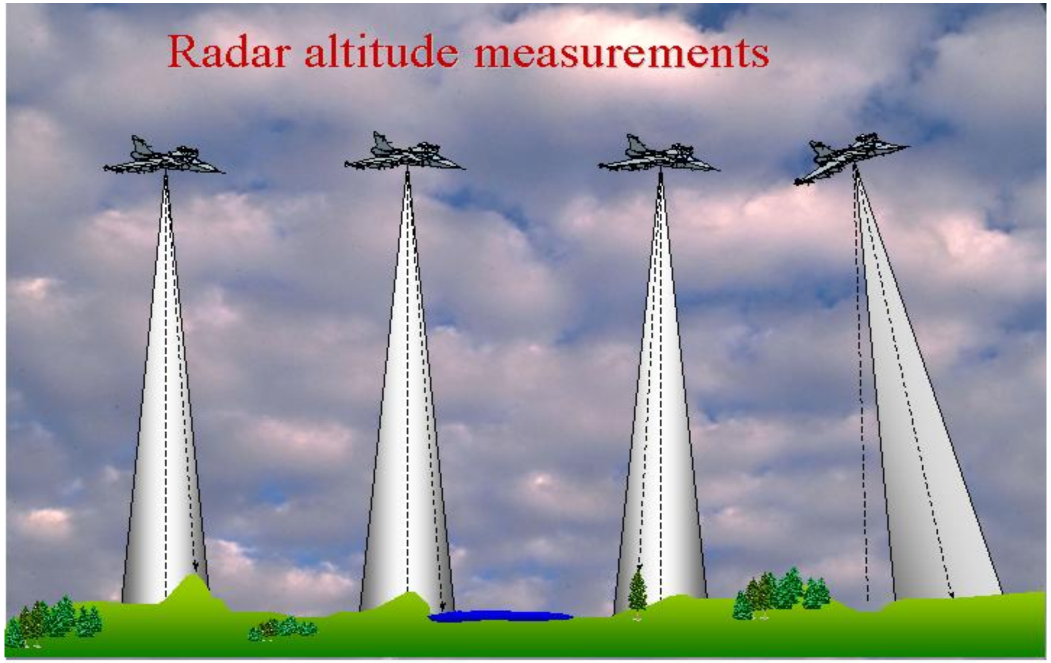
3. Helicopter pose estimation using a map (III/III)

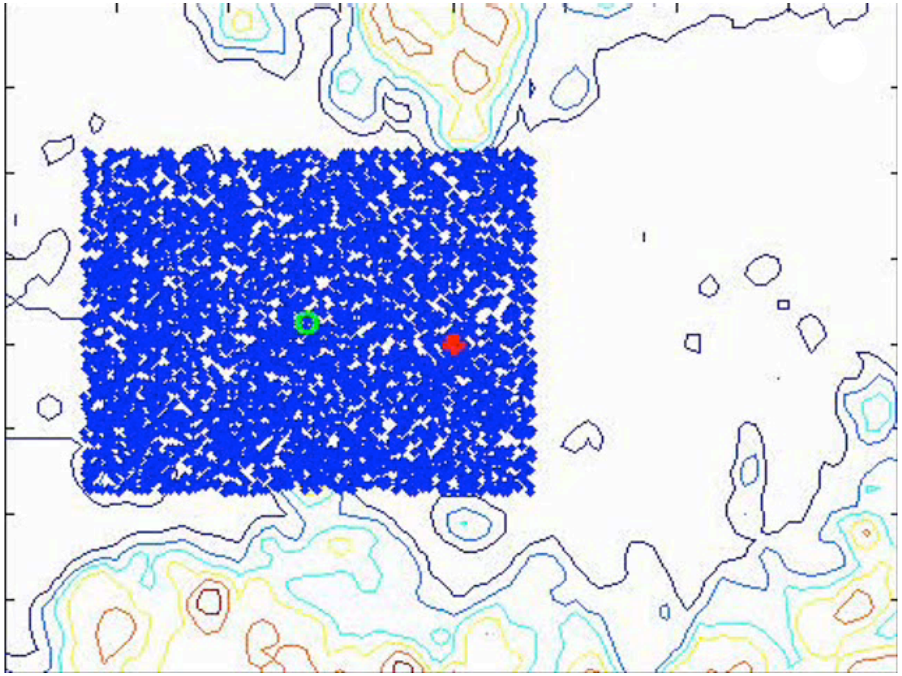


"Think of each particle as one simulation of the system state (in the movie, only the horizontal position is visualized). Only keep the good ones."

Fredrik Lindsten, Jonas Callmer, Henrik Ohlsson, David Törnqvist, Thomas B. Schön, Fredrik Gustafsson, **Geo-referencing for UAV Navigation using Environmental Classification**. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010.

4. Fighter aircraft navigation





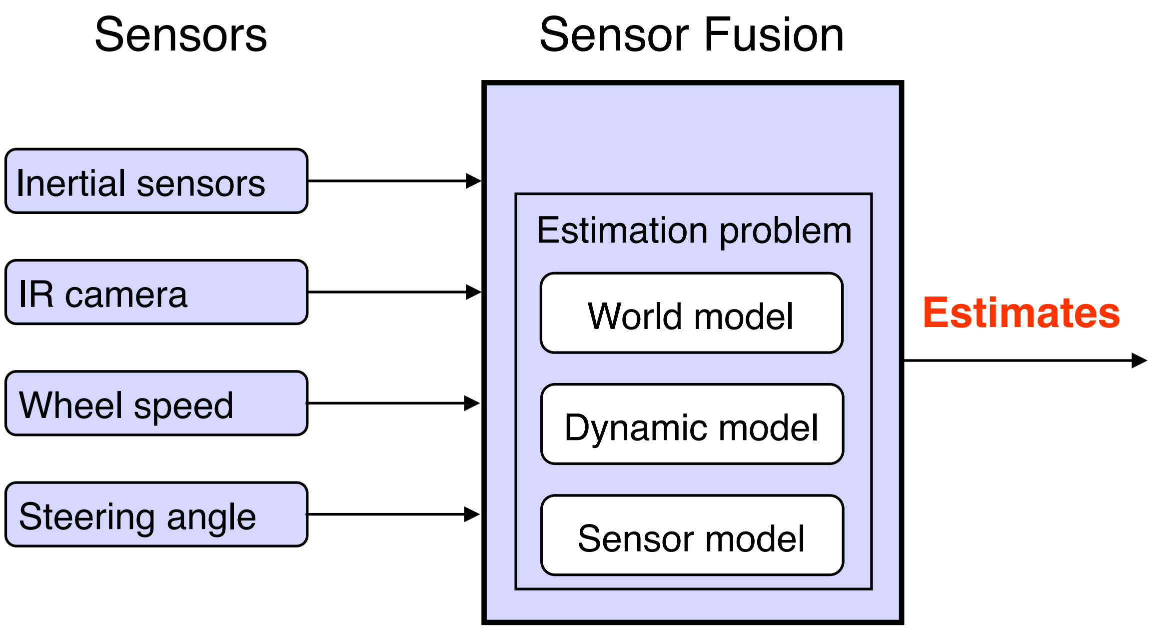
"Think of each particle as one simulation of the system state (in the movie, only the horizontal position is visualized). Only keep the good ones."

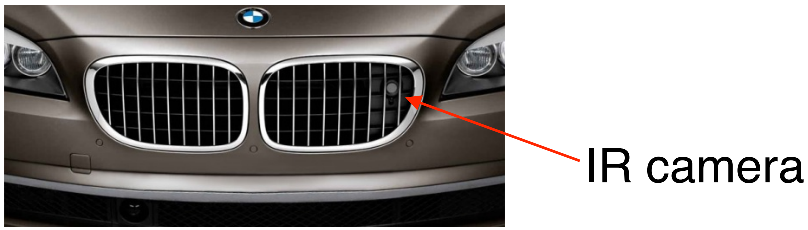
Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund. **Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models**. *IEEE Transactions on Signal Processing*, 53(7):2279-2289, July 2005.

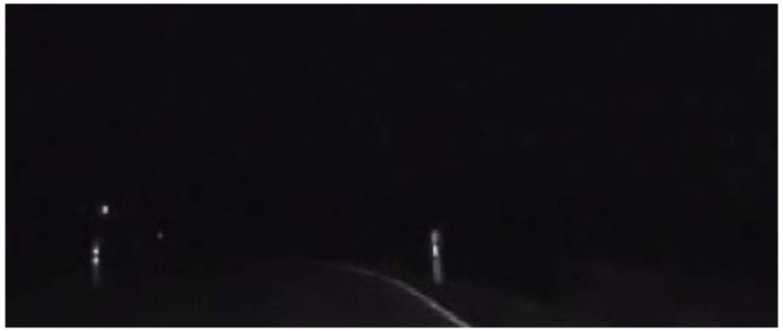
5. Vehicle motion using night vision

Aim: Use images from an infrared (IR) camera in order to obtain better estimates of the ego-vehicle motion and the road geometry in 3D.

Industrial partner: Autoliv





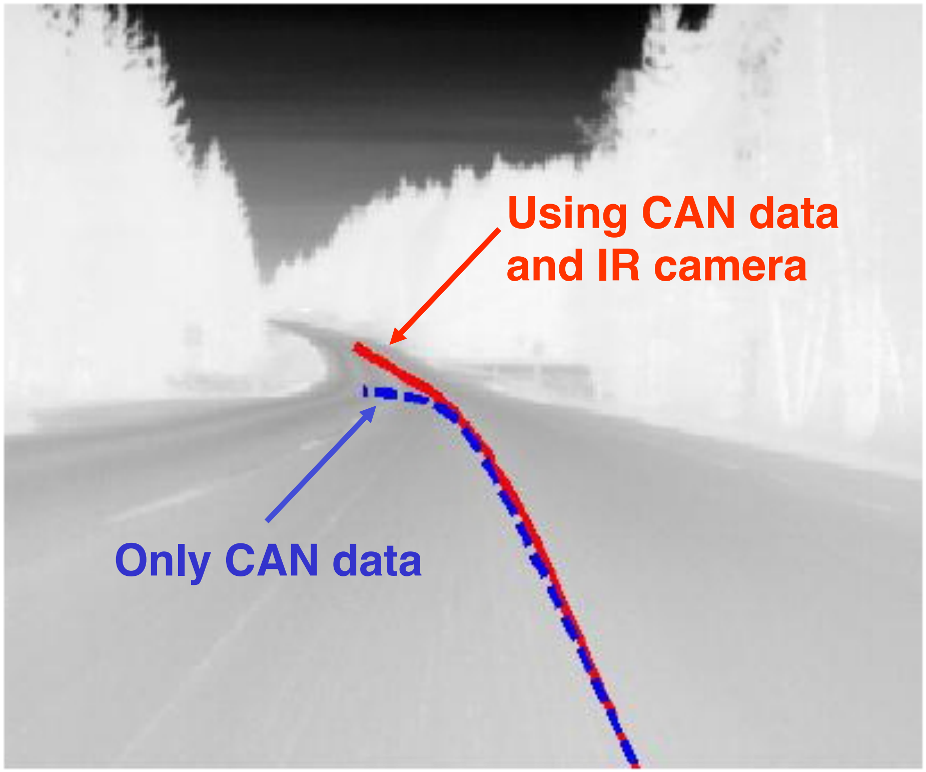


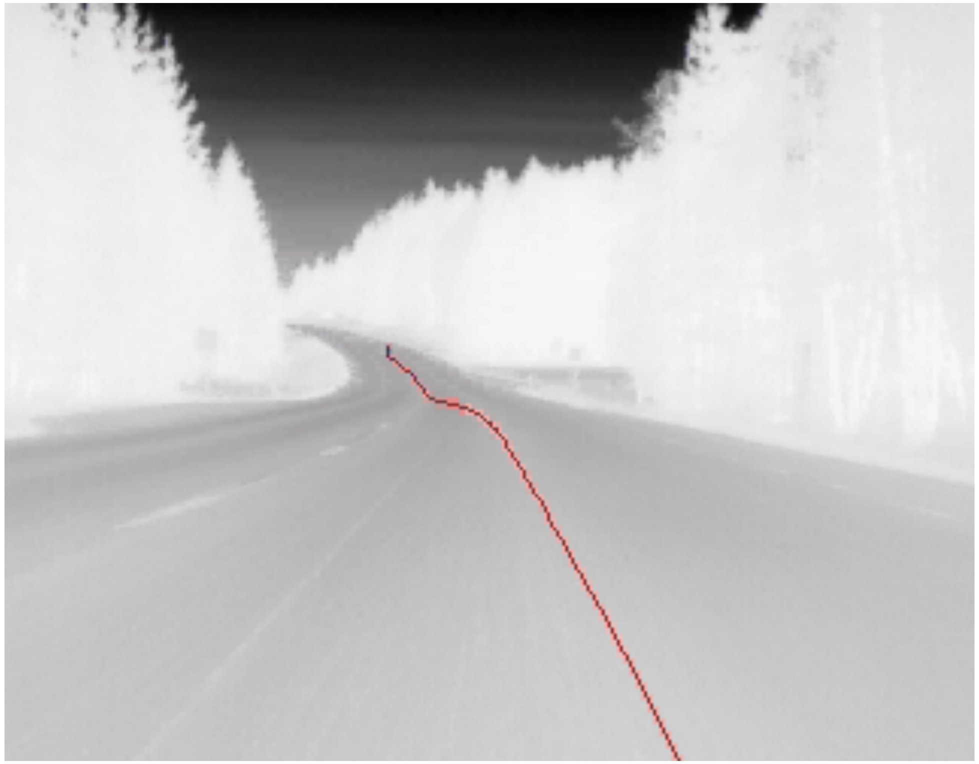
(a) Road scene, as seen with a standard camera.

(b) Same road scene as above, seen with the FIR camera.

5. Vehicle motion using night vision

Measurements recorded during night-time driving on rural roads in Sweden.



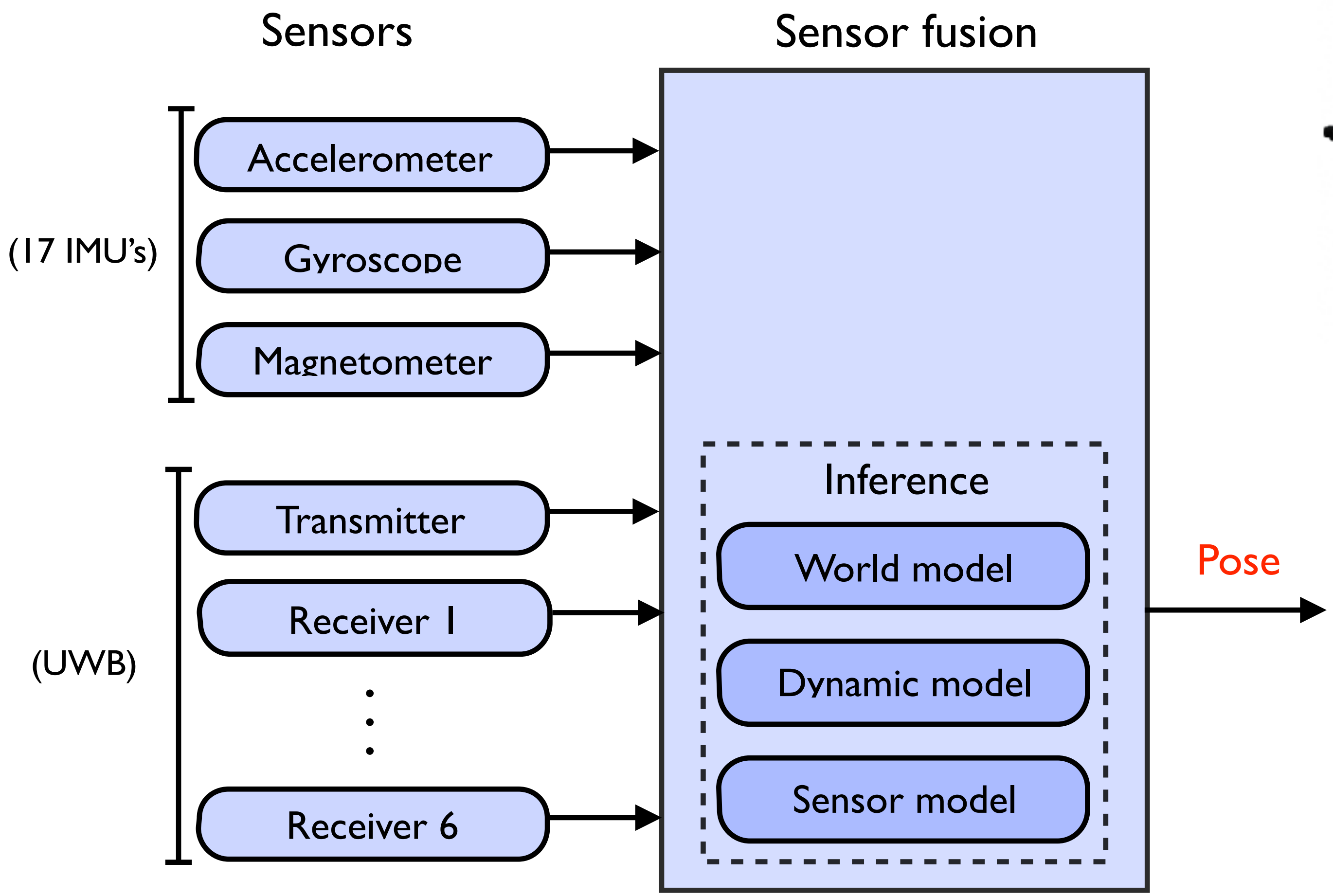


Showing the ego-motion estimates reprojected onto the images.

Thomas B. Schön and Jacob Roll, **Ego-Motion and Indirect Road Geometry Estimation Using Night Vision**. *Proceedings of the IEEE Intelligent Vehicle Symposium (IV)*, Xi' an, China, June 2009.

Aim: Estimate the position and orientation of a human (i.e. human motion) using measurements from inertial sensors and ultra-wideband (UWB).

Industrial partner: Xsens Technologies



6. Indoor human motion estimation (II/V)

Sensor unit integrating an IMU and a UWB transmitter into a single housing.

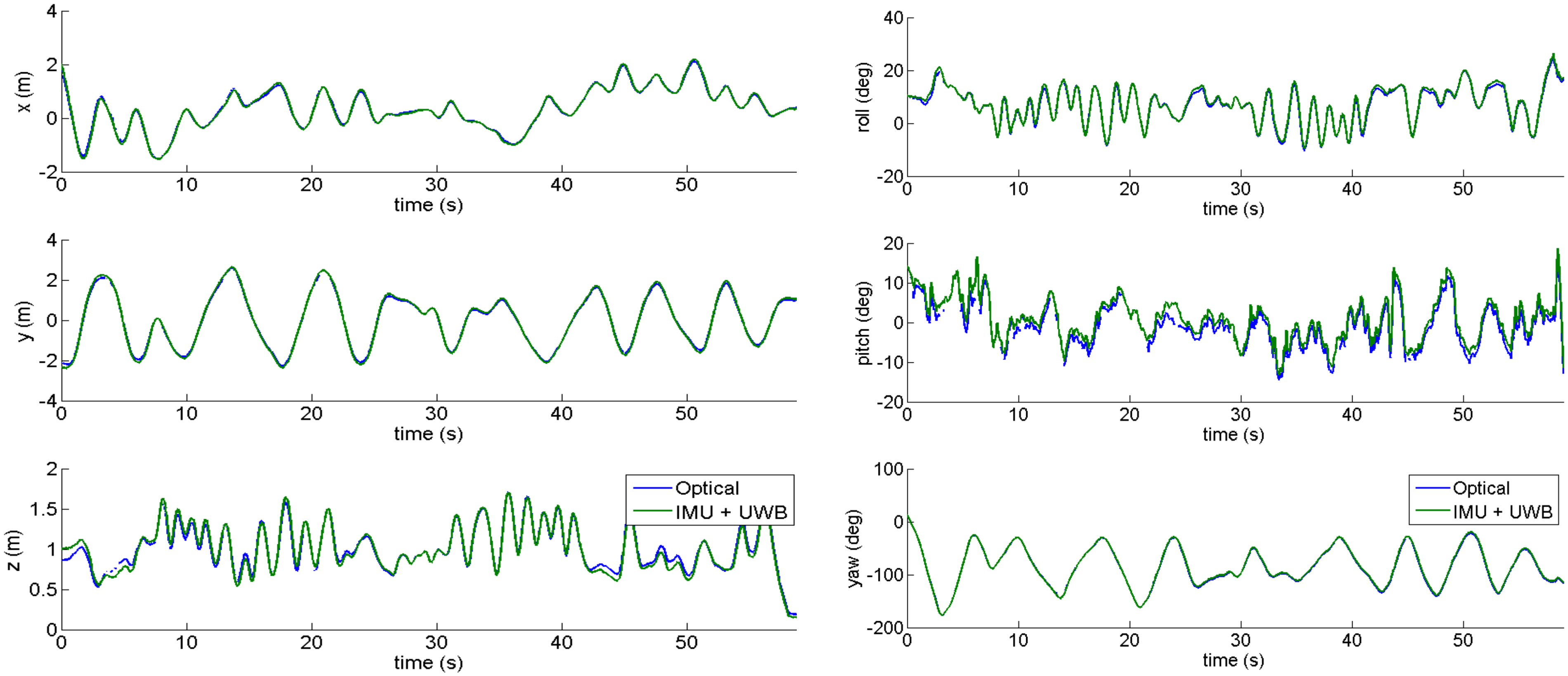
UWB - impulse radio using very short pulses (~ 1ns)

- Low energy over a wide frequency band
- High spatial resolution
- Time-of-arrival (TOA) measurements
- Mobile transmitter and 6 stationary, synchronized receivers at known positions.

Excellent for indoor positioning

- Inertial measurements @ 200 Hz
- UWB measurements @ 50 Hz

6. Indoor human motion estimation (III/V)



Performance evaluation using a camera-based reference system (Vicon).

RMSE: 0.6 deg. in orientation and 5 cm in position.

Manon Kok, Jeroen D. Hol and Thomas B. Schön. **Indoor positioning using ultrawideband and inertial measurements.** *IEEE Transactions on Vehicular Technology*, 64(4):1293-1303, 2015.

Show movie using VLC…

Manon Kok, Jeroen Hol and Thomas B. Schön. **An optimization-based approach to human body motion capture using inertial sensors**. I*n Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC)*, Cape Town, South Africa, August 2014.

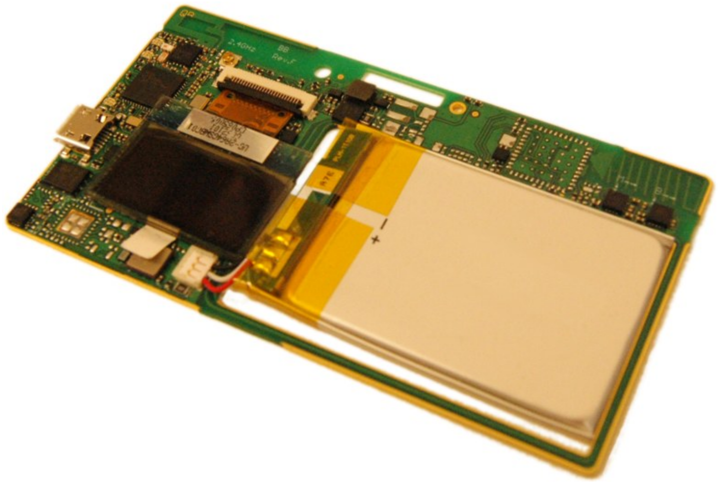
Show movie using VLC…

Manon Kok, Jeroen Hol and Thomas B. Schön. **An optimization-based approach to human body motion capture using inertial sensors**. I*n Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC)*, Cape Town, South Africa, August 2014.

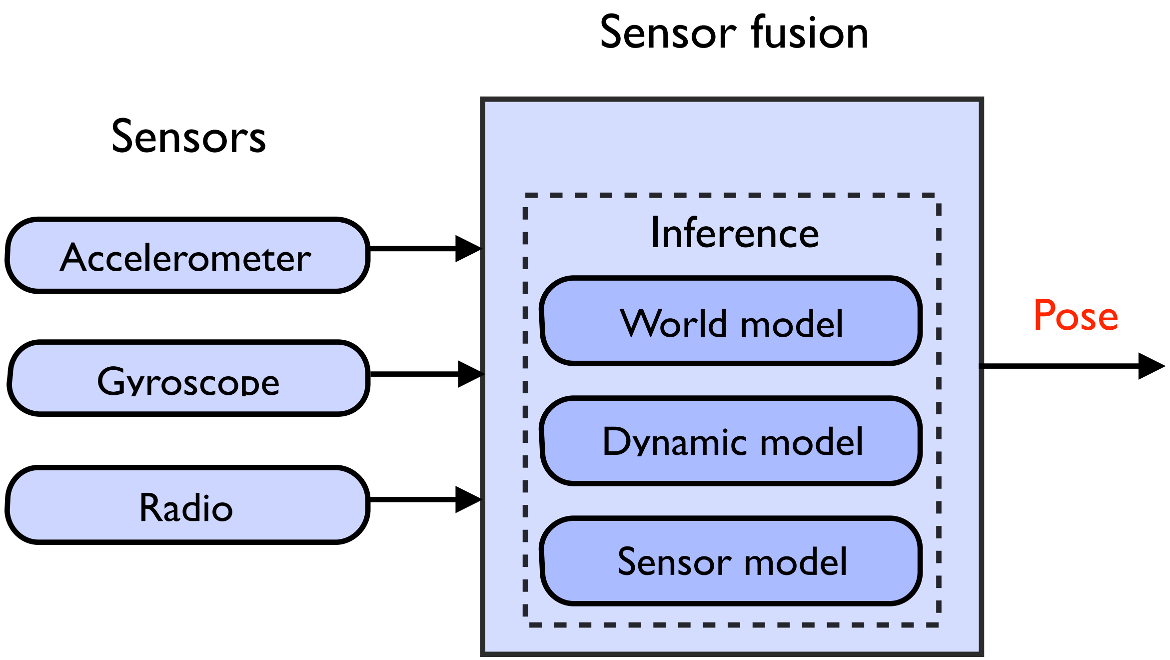
7. Indoor positioning of humans (I/III)

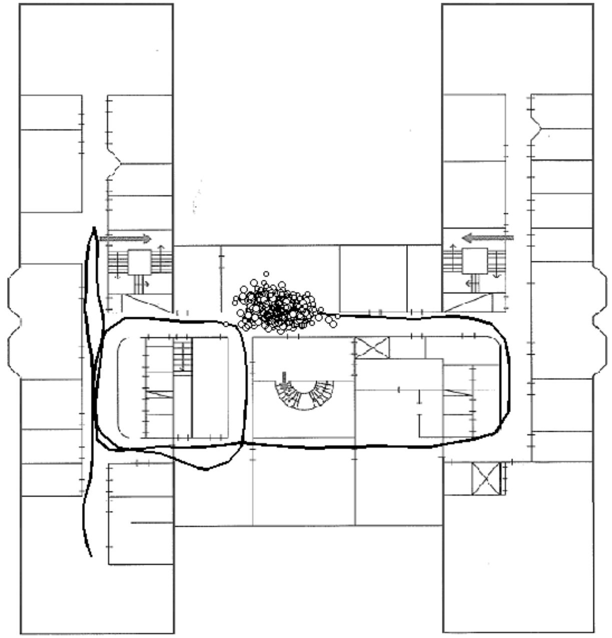
Aim: Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.

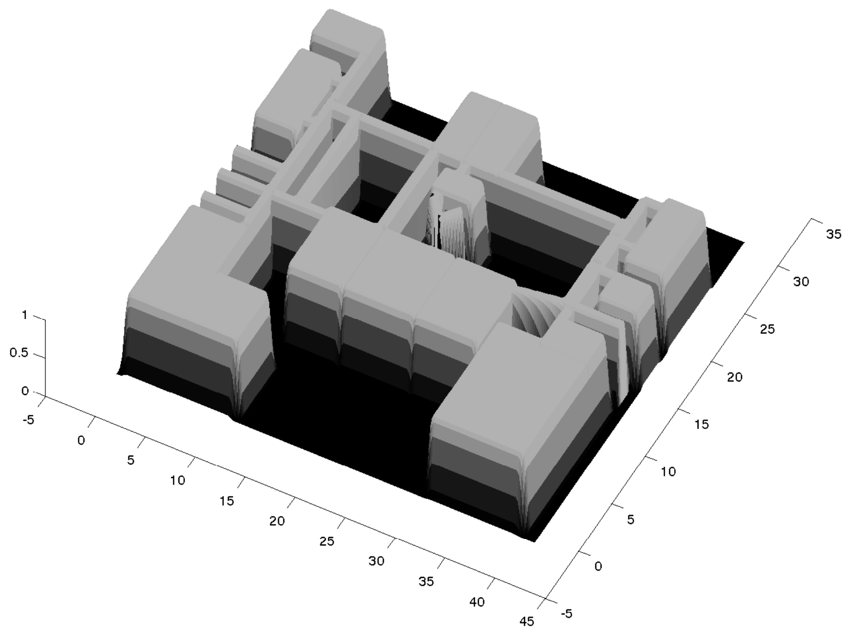
Industrial partner: Xdin



7. Indoor positioning of humans (II/III)

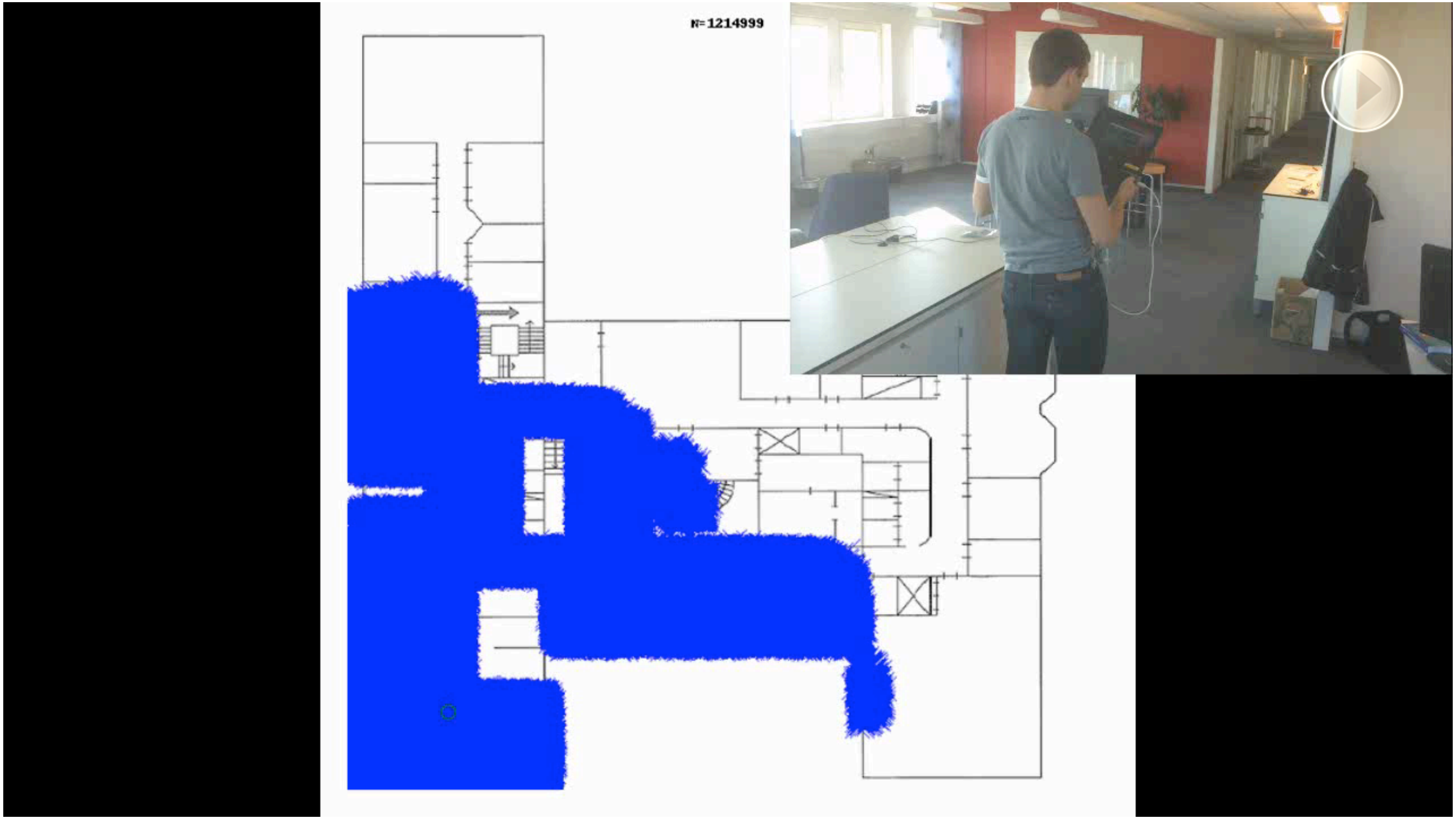






PDF of an office environment, the bright areas are rooms and corridors (i.e., walkable space).

7. Indoor positioning of humans (III/III)



Conclusions

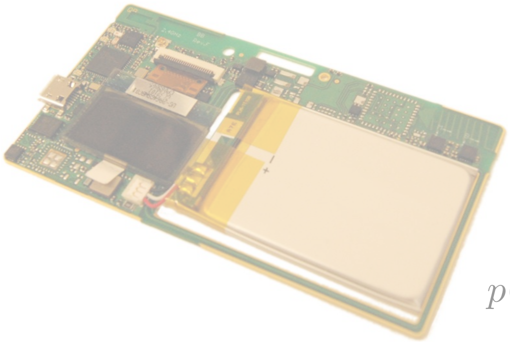
Quite a few different applications from different areas, all solved using the
same underlying system identification and sensor fusion strategy

- **Model** the dynamics (possibly sys.id.)
- **Model** the sensors (possibly sys.id.)
- **Model** the world
- Solve the resulting **inference** problem

and, do not underestimate the "surrounding infrastructure" (possibly sys.id.)!

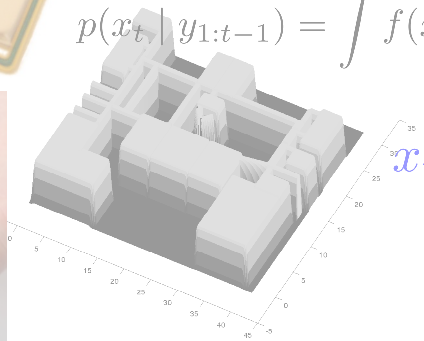
- There is a lot of **interesting research** that remains to be done!
- The number of available sensors is currently skyrocketing
- The **industrial utility** of this technology is **growing** as we speak!

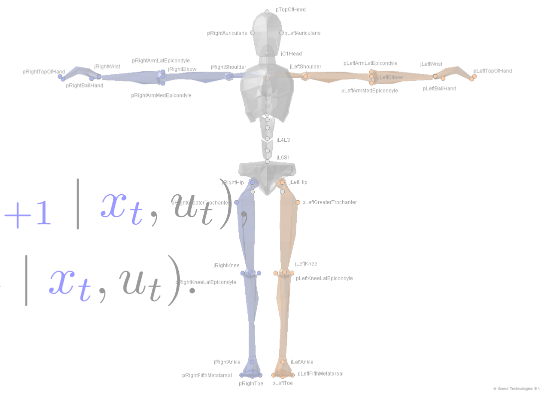
Thank you for your attention!!



$$p(x_t \mid y_{1:t}) = \frac{h(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})},$$

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1})p(x_{t-1} \mid y_{1:}$$





$$x_{t+1} \mid x_t \sim f_\theta(x_{t+1} \mid x_t, u_t),$$
$$y_t \mid x_t \sim h_\theta(y_t \mid x_t, u_t).$$