EPFL

Disal

## Signals, Instruments, and Systems – W11

# Environmental Sensing Systems – Robotic Localization in Practice and Sensor Networks

# Outline

- Localization in mobile robotics
  - The Extended Kalman Filter algorithm in multiple dimensions
  - A practical recipe for localization with Kalman filters in mobile robotics

- Environmental sensor networks
  - Motivation
  - Examples of sensor networks
  - The Sensorscope project

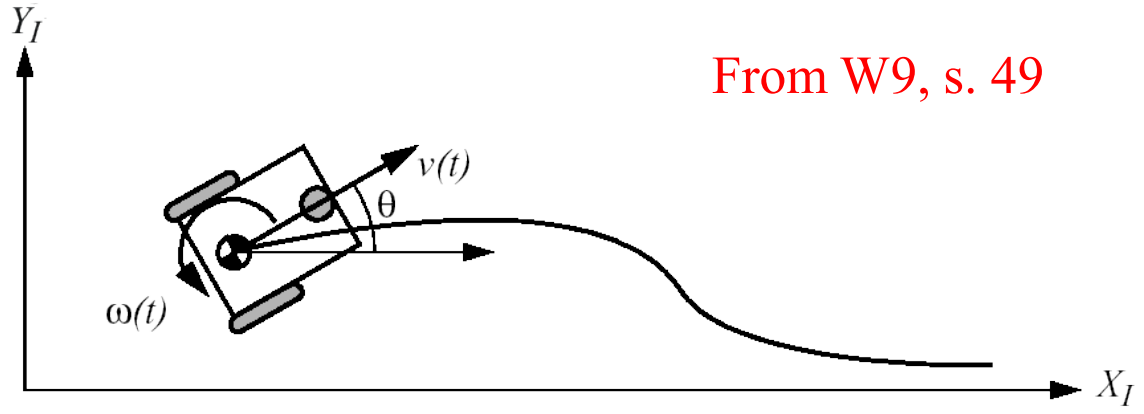
# The Extended Kalman Filter Algorithm for Multi-Dimension Problems

# Pose Variation During Δt

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

$$\begin{cases} \Delta x = \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta y = \Delta s \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \end{cases}$$

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \xrightarrow{t'=t+\Delta t} p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix}$$




From W9, s. 49

b = inter-wheel distance
$\Delta s_r$ = traveled distance right wheel
$\Delta s_l$ = traveled distance left wheel
$\Delta\theta$ = orientation change of the vehicle

$f_{1,2}$: nonlinear!

$$\Delta p = p' - p = f(\Delta s_r, \Delta s_l) = \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} = \begin{bmatrix} f_1(\Delta s_r, \Delta s_l) \\ f_2(\Delta s_r, \Delta s_l) \\ f_3(\Delta s_r, \Delta s_l) \end{bmatrix}$$

# Linear vs. Nonlinear System and Measurement

Linear system equation: $\boldsymbol{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$

Linear measurement equation: $\boldsymbol{z_t = C_t x_t + \delta_t}$

Nonlinear system equation: $\boldsymbol{x_t = f(x_{t-1}, u_t) + \varepsilon_t}$

Nonlinear measurement equation: $\boldsymbol{z_t = h(x_t) + \delta_t}$

Linearization via first order Taylor series expansion!

[Adapted from Thrun et al., 2005]

# Kalman Filter Algorithm

Algorithm **Kalman_filter**($\boldsymbol{x_{t-1}}$, $\Sigma_{t-1}$, $\boldsymbol{u_t}$, $\boldsymbol{z_t}$)

Prediction:

1. $\hat{\boldsymbol{x}}_{\boldsymbol{t}} = A_t \boldsymbol{x}_{\boldsymbol{t-1}} + B_t \boldsymbol{u}_{\boldsymbol{t}}$
2. $\hat{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

Correction or update:

3. $K_t = \hat{\Sigma}_t C_t^T (C_t \hat{\Sigma}_t C_t^T + Q_t)^{-1}$
4. $\boldsymbol{x}_{\boldsymbol{t}} = \hat{\boldsymbol{x}}_{\boldsymbol{t}} + K_t(\boldsymbol{z}_{\boldsymbol{t}} - C_t \hat{\boldsymbol{x}}_{\boldsymbol{t}})$
5. $\Sigma_t = (I - K_t C_t)\hat{\Sigma}_t$

Return $\boldsymbol{x}_{\boldsymbol{t}}$, $\Sigma_t$

[Adapted from Thrun et al., 2005]

# Extended Kalman Filter Algorithm

Relevant Jacobians:

$$F_t(\boldsymbol{x_{t-1}}, \boldsymbol{u_t}) = \frac{\partial f(\boldsymbol{x_{t-1}}, \boldsymbol{u_t})}{\partial \boldsymbol{x_{t-1}}}$$

$$H_t(\hat{\boldsymbol{x}}_t) = \frac{\partial h(\hat{\boldsymbol{x}}_t)}{\partial \boldsymbol{x_t}}$$

(see s. 51-52, W9 for Jacobian examples)

Algorithm **EKF** ($\boldsymbol{x_{t-1}}$, $\Sigma_{t-1}$, $\boldsymbol{u_t}$, $\boldsymbol{z_t}$)

Prediction

1. $\hat{\boldsymbol{x}}_t = f(\boldsymbol{x_{t-1}}, \boldsymbol{u_t})$
2. $\hat{\Sigma}_t = F_t \Sigma_{t-1} F_t^T + R_t$

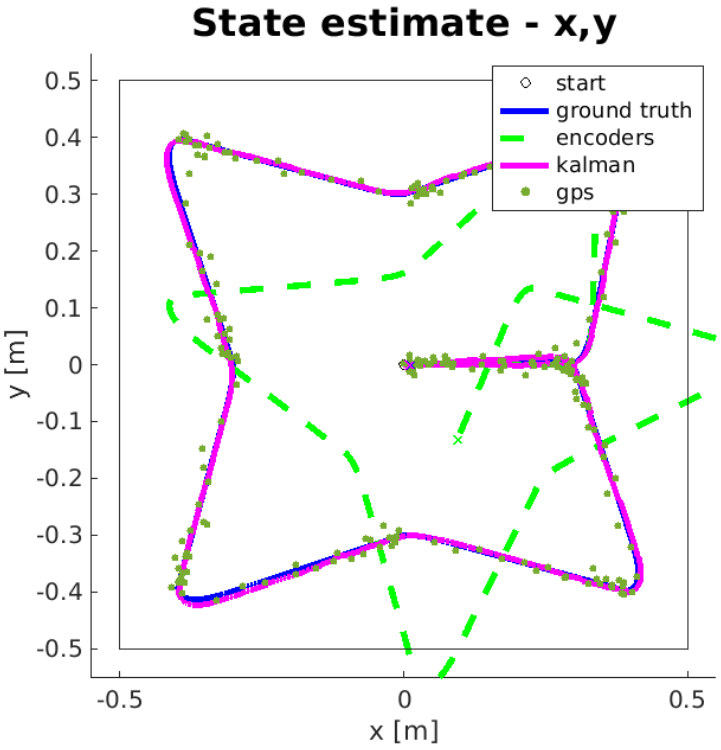
Correction or update

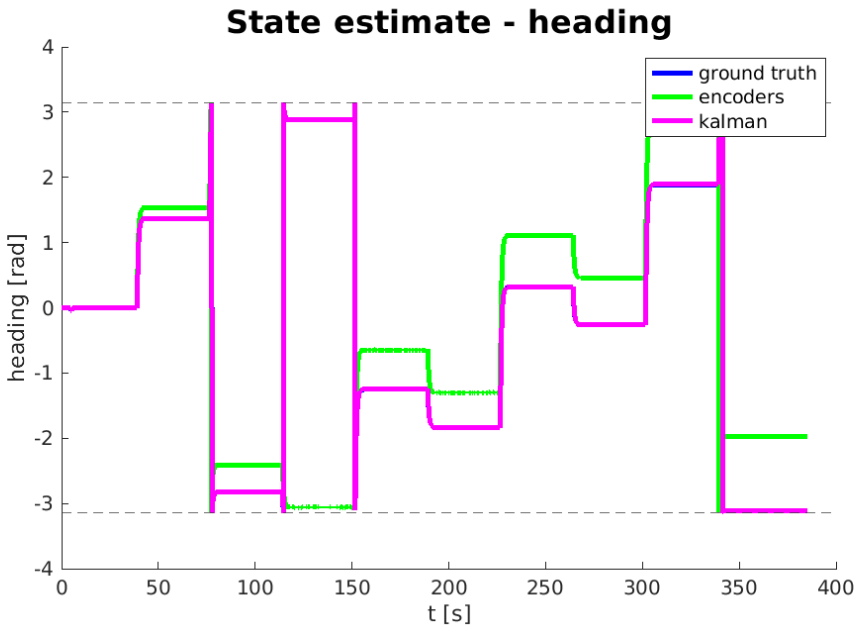
3. $K_t = \hat{\Sigma}_t H_t^T (H_t \hat{\Sigma}_t H_t^T + Q_t)^{-1}$
4. $\boldsymbol{x_t} = \hat{\boldsymbol{x}}_t + K_t(\boldsymbol{z_t} - h(\hat{\boldsymbol{x}}_t))$
5. $\Sigma_t = (I - K_t H_t)\hat{\Sigma}_t$

Return $\boldsymbol{x_t}$, $\Sigma_t$

[Adapted from Thrun et al., 2005]

# Example of Results using EKF

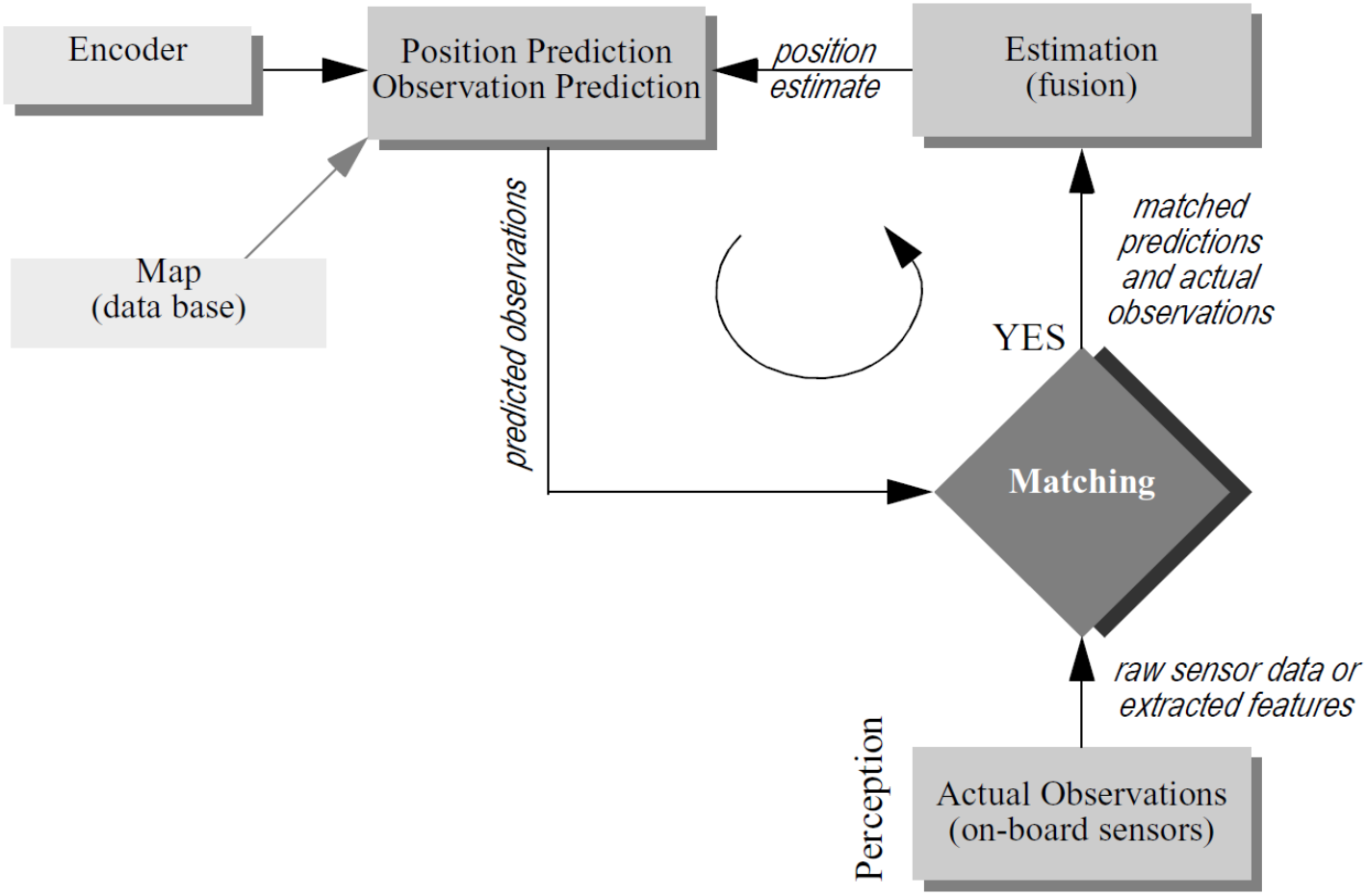



Fusion of encoder-based odometry with GPS in high-fidelity simulation

# A Practical Recipe for Localization with Kalman Filters in Mobile Robotics

# Localization with a Kalman Filter



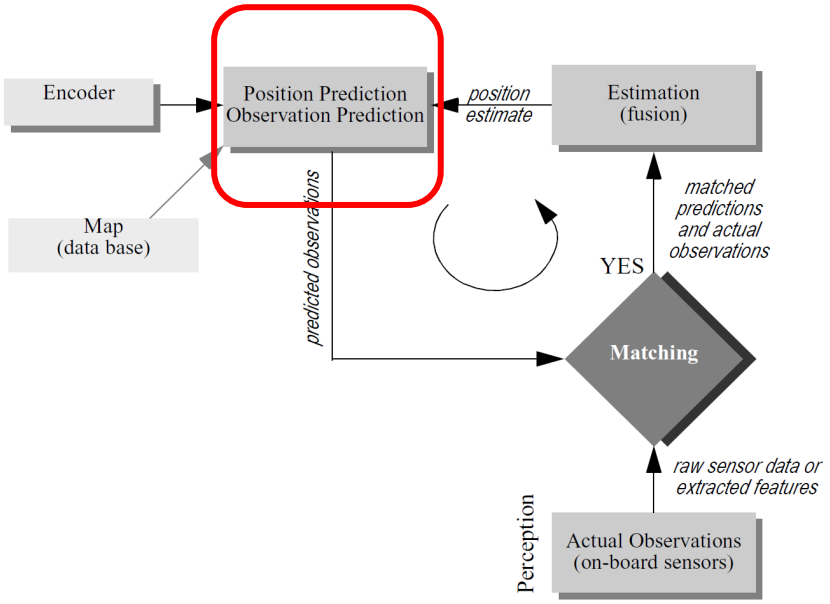

[From Siegwart and Nourbakhsh, 2004]

# A Five-Step Iterative Recipe

1. Robot pose prediction
2. Actual observations
3. Predicted observations
4. Matching
5. Estimation with Extended Kalman Filter

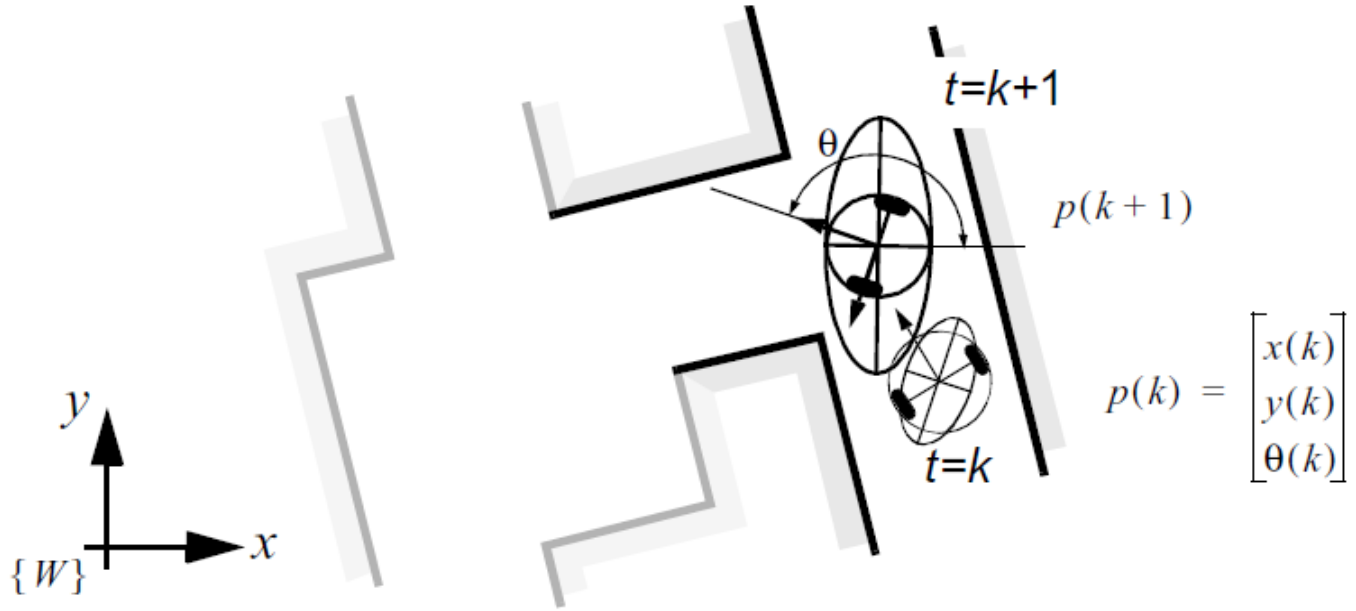
# Robot Pose Prediction

- Input: motion model of the vehicle, odometry, control actions
- Output: estimation mean and co-variance of the pose at the next timestep



# Robot Pose Prediction - Example

- Differential-drive wheeled robot (e.g., e-puck, Pioneer 3-AT)
- Laser range finder (i.e. LIDAR) as exteroceptive sensor
- Map of a built environment with lines



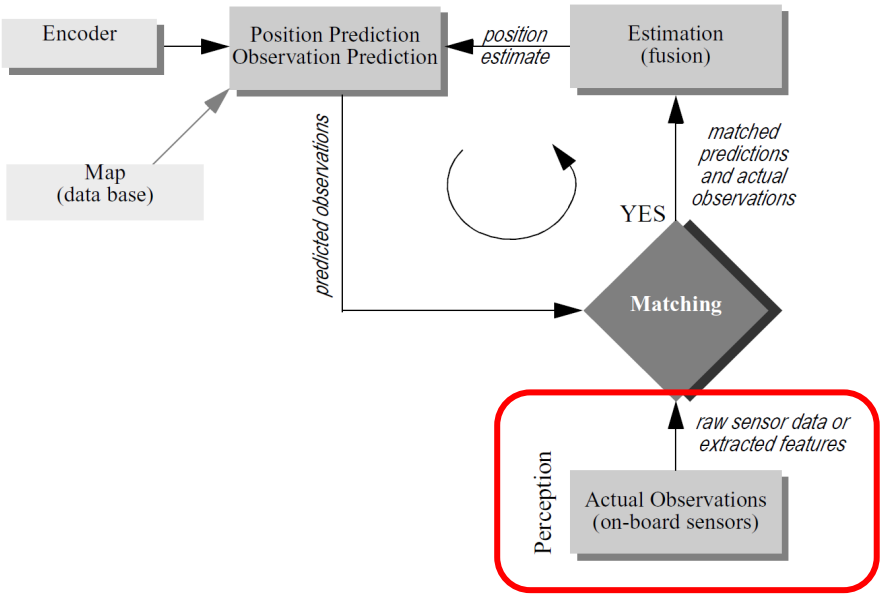

[From Siegwart and Nourbakhsh, 2004]

# Actual Observations

- Input: sensor model

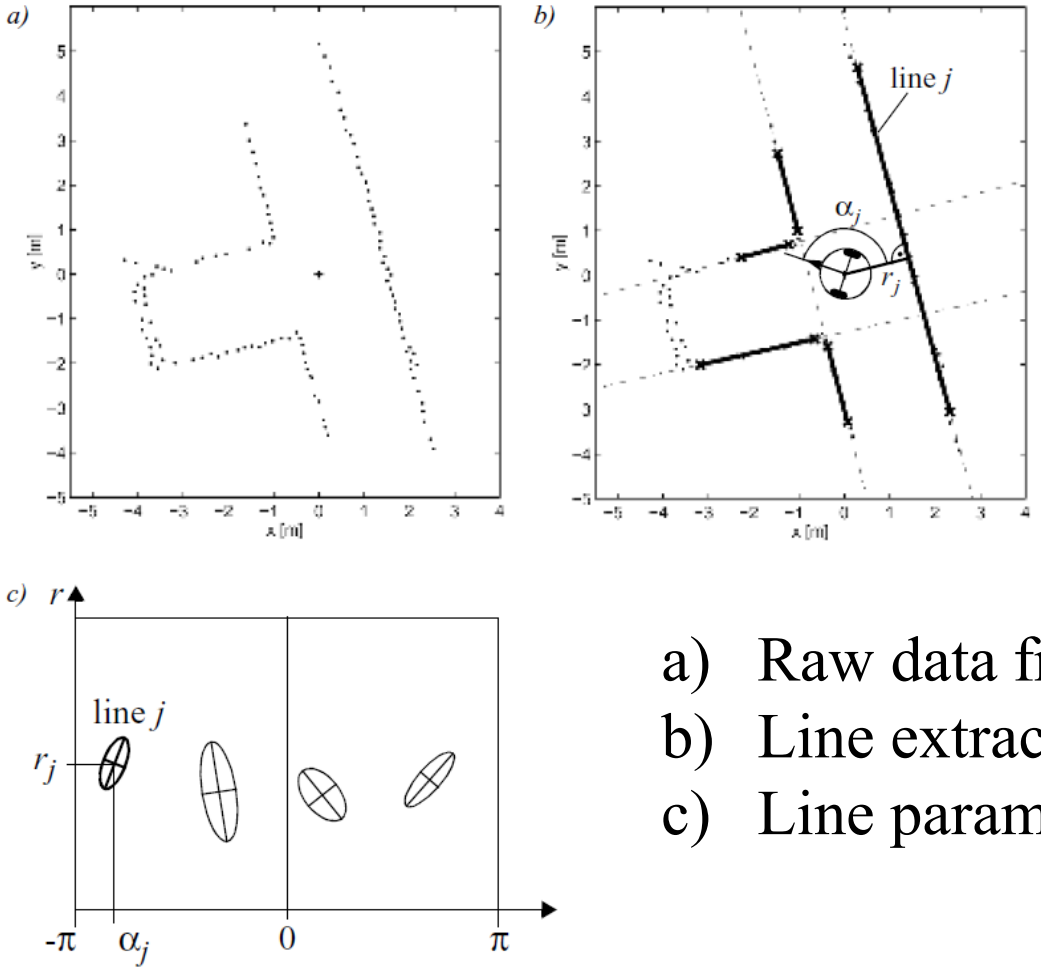
  Coordinate transformation needed for getting to the local frame (in this case sensor frame)

- Output: depending on the sensing technology and feature (e.g., distance, line, door)



# Actual Observations – Example

[From Siegwart and Nourbakhsh, 2004]



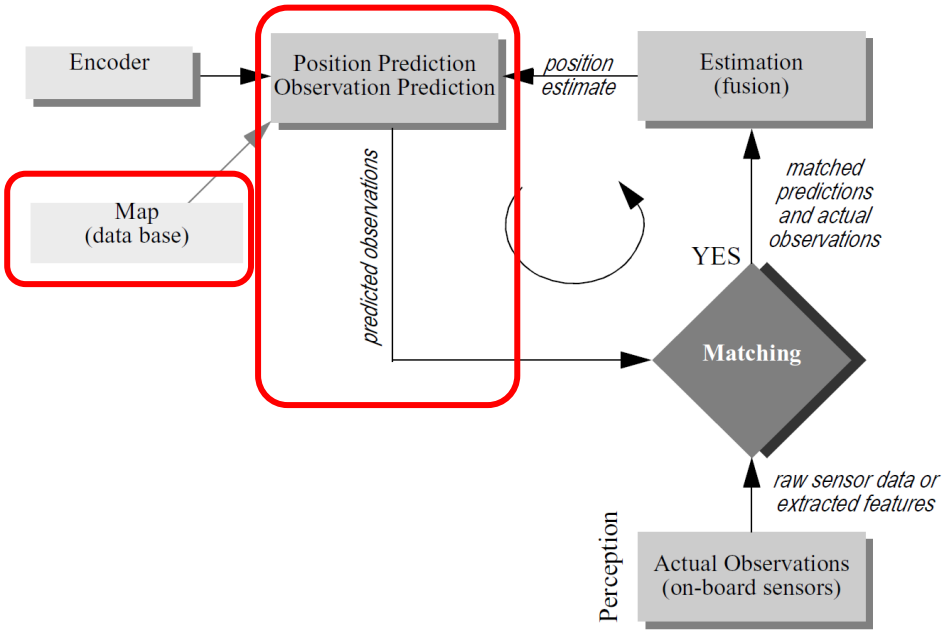

a) Raw data from the LIDAR
b) Line extraction (see also s. 5, W10)
c) Line parametrization with uncertainties

# Predicted Observations

- Input: predicted robot pose and map

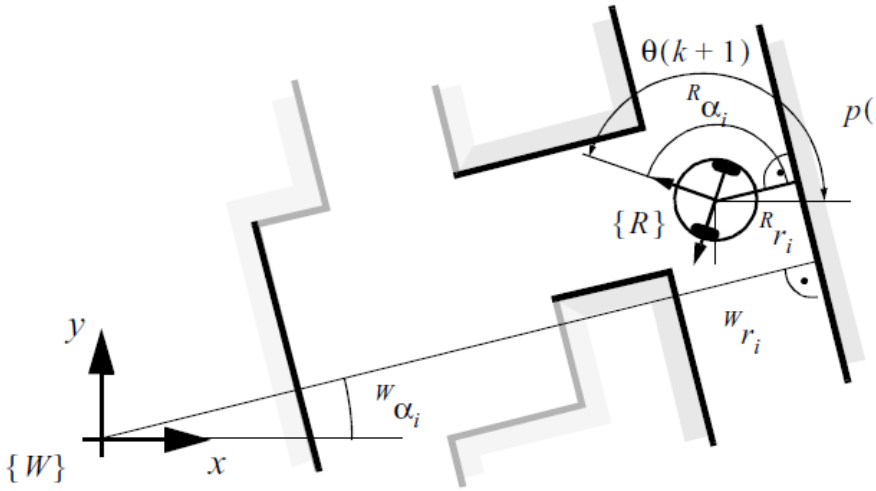
  Coordinate transformation needed for getting to the local frame (in this case sensor frame, in the example assumed to be the same of the robot frame)

- Output: predicted feature observations



# Predicted Observations - Example

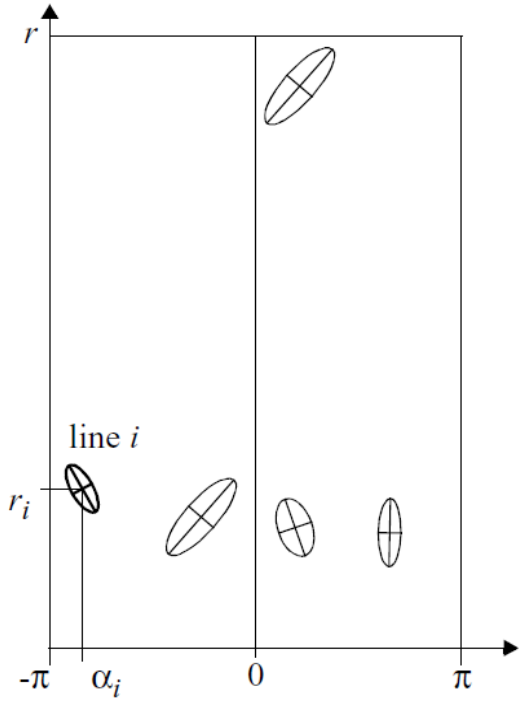
[From Siegwart and Nourbakhsh, 2004]




$$p(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1)) \end{bmatrix}$$

{R}: Robot (body) frame

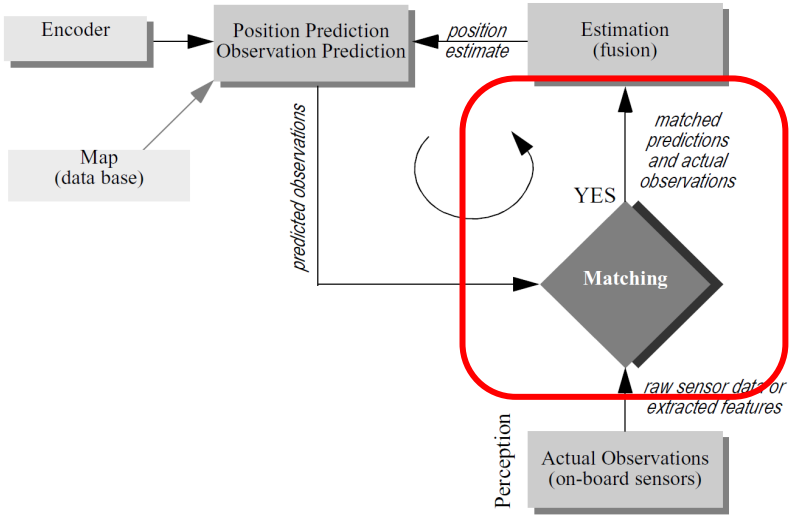
{W}: World (inertial) frame




Measurement prediction (line model space)

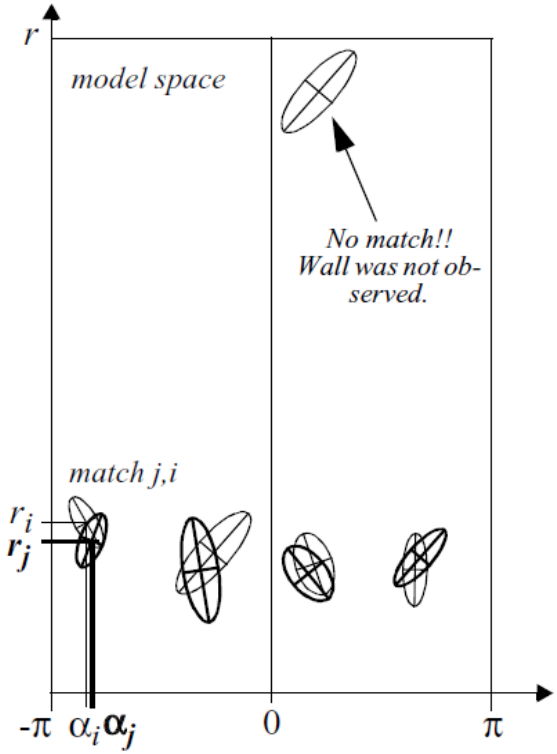
# Matching

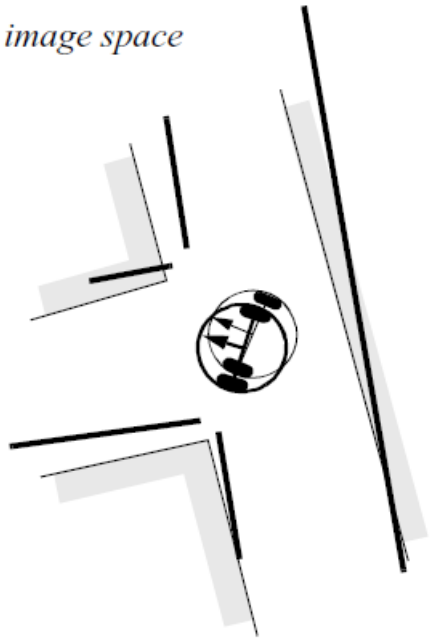
- Input: predicted observations and actual observations, all in the sensor frame
  - Objective: matching actual to predicted observations
  - Innovation metric: difference predicted and observed measurements (also random variable, with mean and co-variance)
  - Validation criterion based on innovation falling into validation gates
- Output: selection of features worth using in the estimation process



# Matching - Example

[From Siegwart and Nourbakhsh, 2004]




Observation (thick)
vs. prediction (thin)
in model space

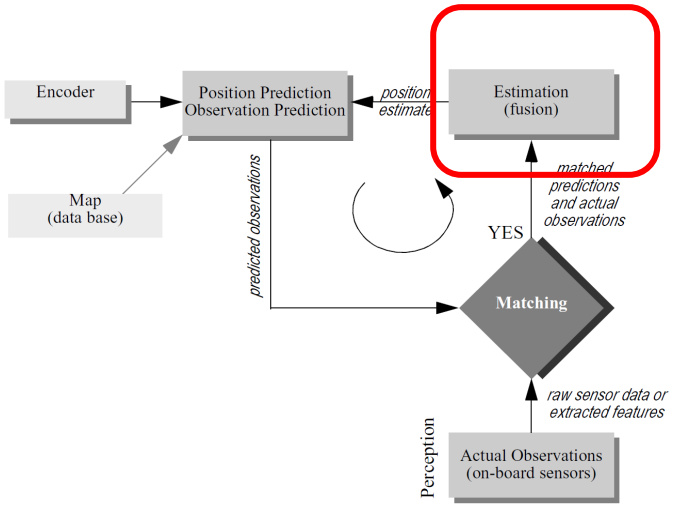
Observation (thick)
vs. prediction (thin)
in image space

# Estimation with Extended Kalman Filter

- Input: pose prediction from the motion model and validated observations

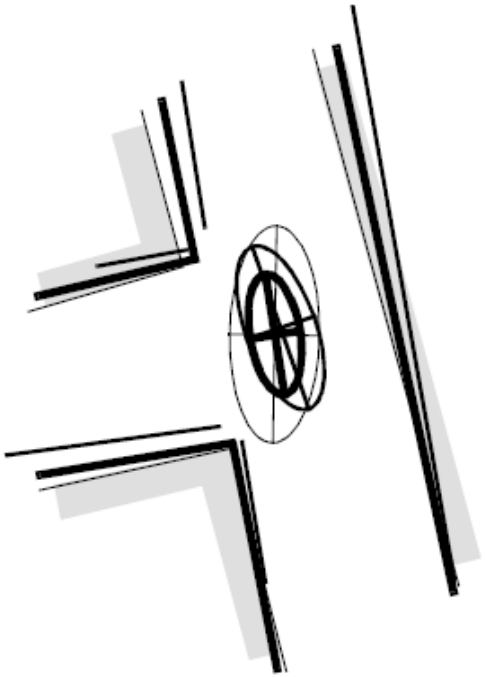
  Objective: fuse pose prediction from motion model and all validated observations to create next pose

- Output: next mean and variance of the pose



# Estimation with EKF - Example

[From Siegwart and Nourbakhsh, 2004]



- Thin: prediction
- Thick: observation
- Very thick: updated (or corrected) result with EKF

# Motivation and Examples of Sensor Networks

# Motivation for Sensor Networks

What if we could monitor events which …

- have a large *spatial* and *temporal* distribution
- require *in-situ* measurements
- take place in hard to access places
- generate data which need to be available in *real-time*

# Motivation for Sensor Networks

What would we need for that?

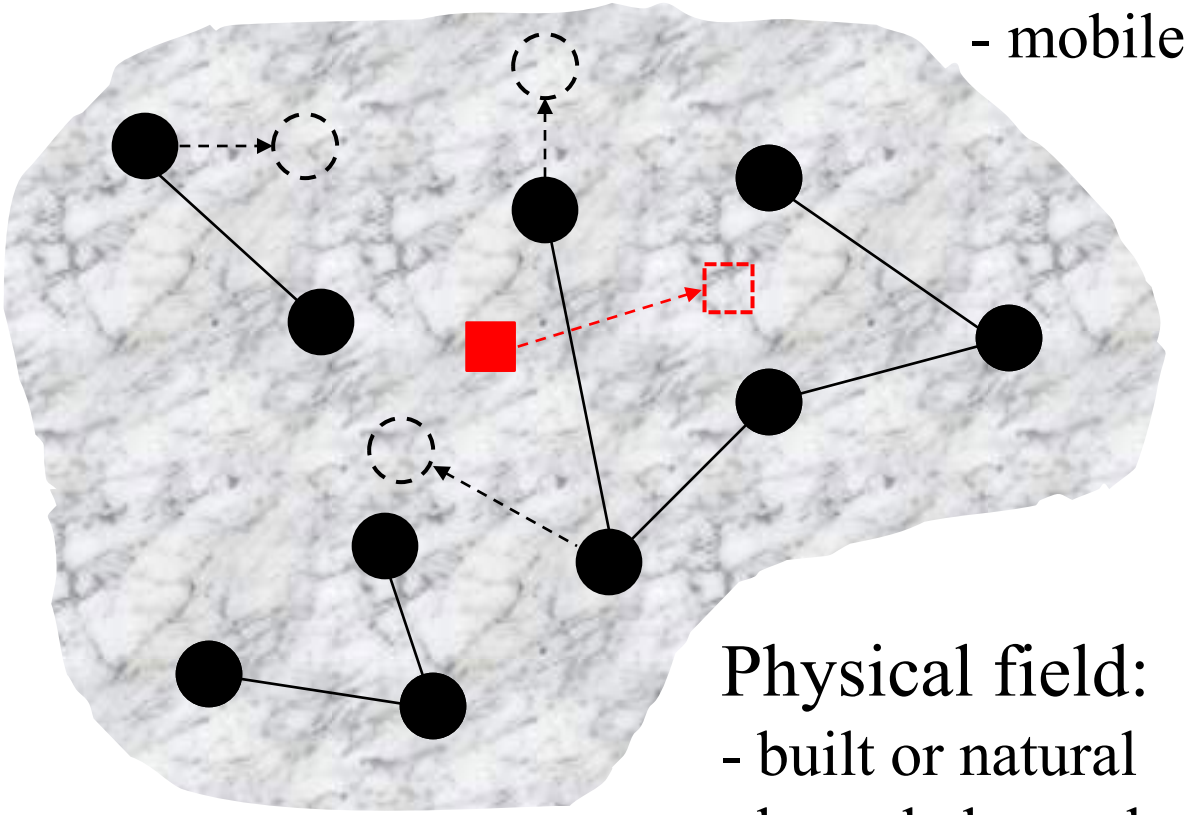
A device which …

- is cheap – so we can distribute *many* of it
- is reliable – so we can measure for a long *time*
- uses little power – battery/solar cell powered
- has a radio – so it can *communicate*
- can potentially move – so it can potentially *relocate*

# Environmental Monitoring

Typical solution in environmental monitoring:
- sparse sensing
- expensive station(s)
- field estimation via physics- or chemistry-based models
- possible mobility

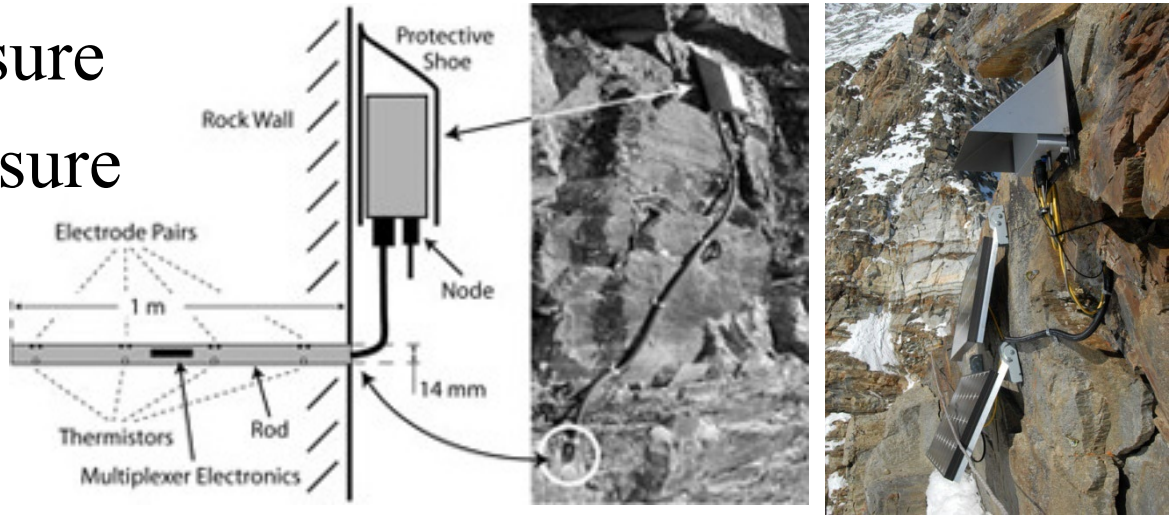
Distributed solution for augmentation:
- size, cost
- number
- networked
- mobile



Physical field:
- built or natural
- bounded or unbounded
- 2D or 3D

# Permafrost Monitoring
## *Permasense*

- What is measured:
  - rock temperature
  - rock resistivity
  - crack width
  - earth pressure
  - water pressure

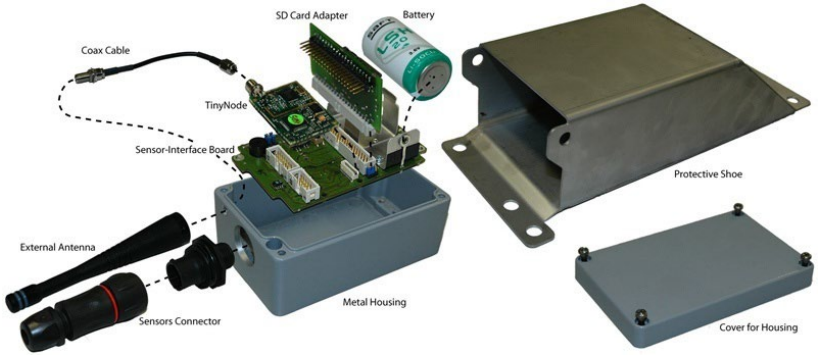



*Pictures: courtesy of Permasense*

# Permafrost Monitoring
## *Permasense*

- Why:

  "[…] gathering of environmental data that helps to understand the processes that connect climate change and rock fall in permafrost areas"

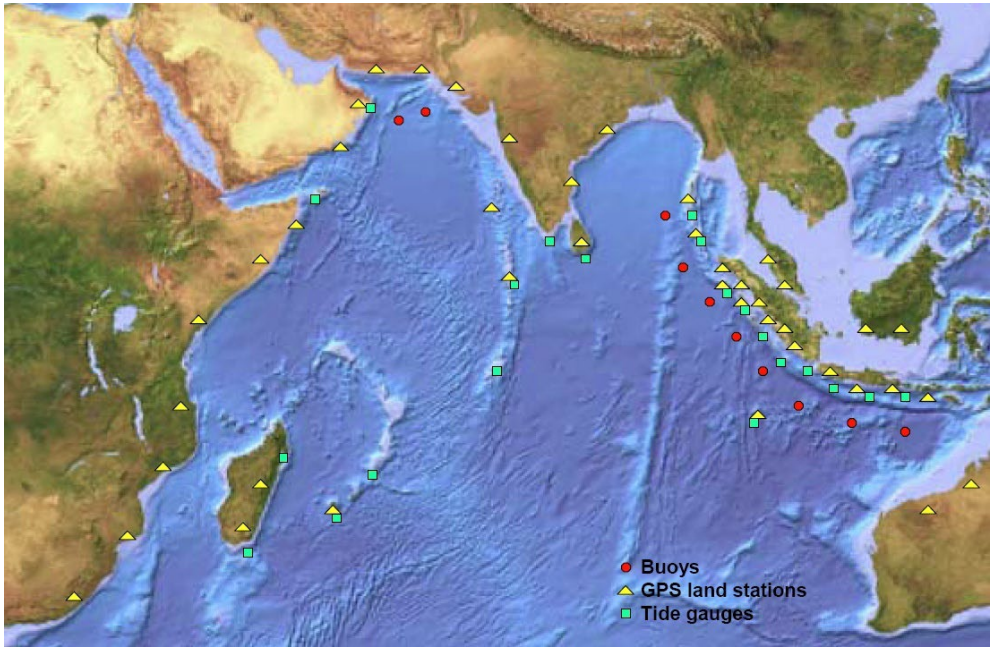




*Pictures: courtesy of Permasense*

# Tsunami Early Warning System *GITEWS*

- German-Indonesian effort
- What is measured:
  - seismic events
  - water pressure

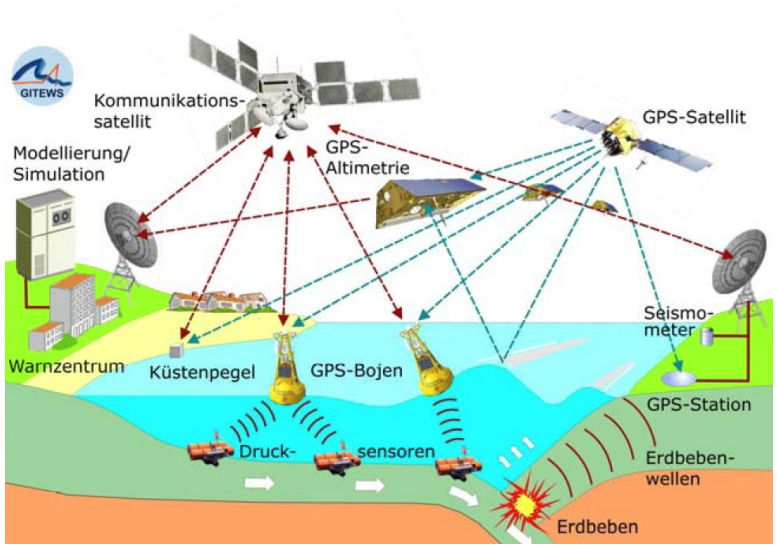



*Pictures: courtesy of Deutsches GeoForschungsZentrum (GFZ)*

# Tsunami Early Warning System *GITEWS*

- Why:

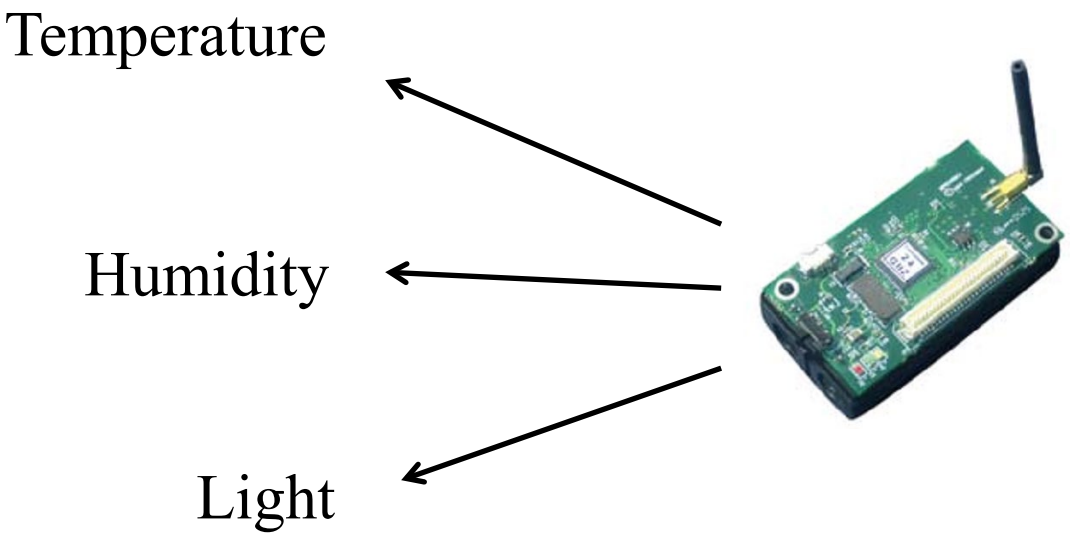
  To detect seismic events which could cause a Tsunami. Detect a Tsunami and predict its propagation.



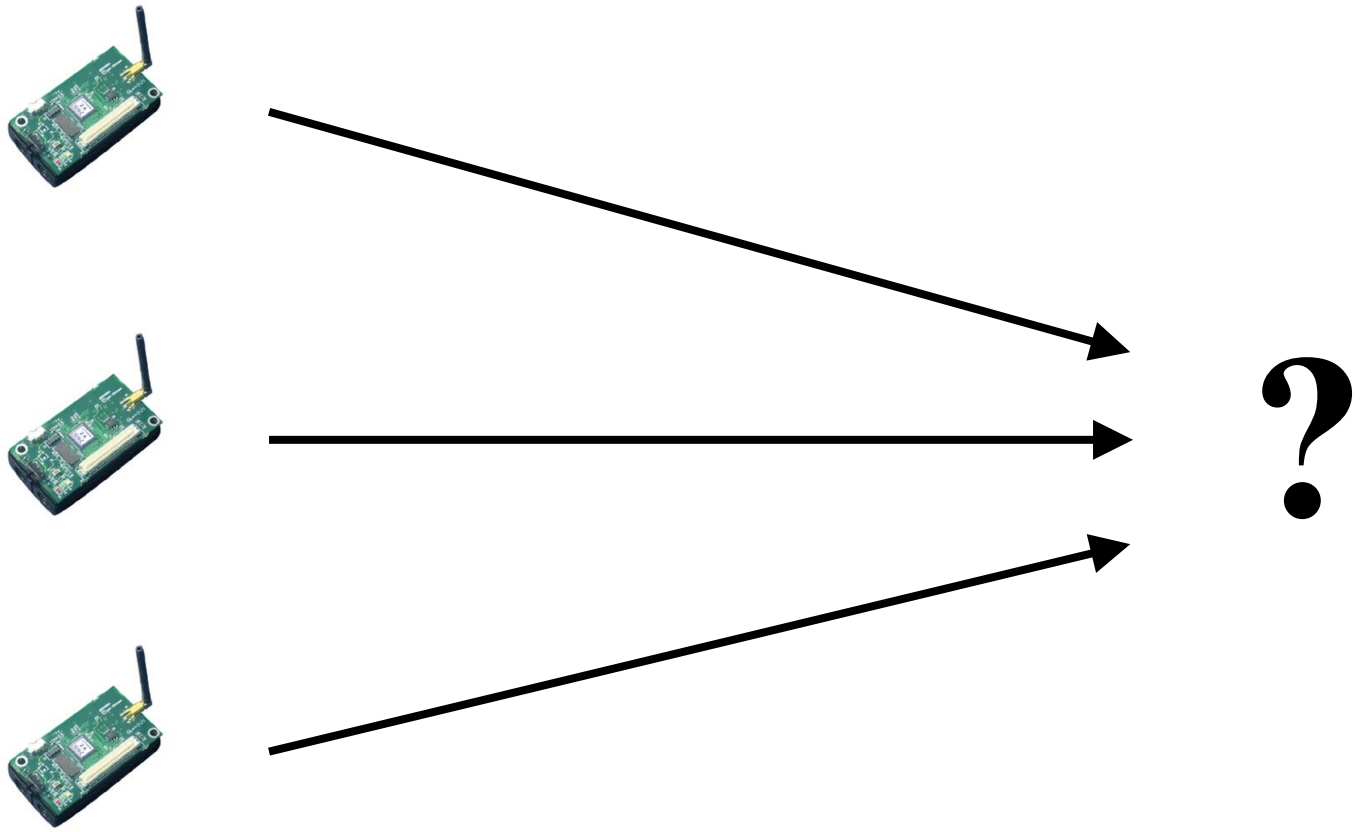

*Pictures: courtesy of Deutsches GeoForschungsZentrum (GFZ)*

# The Sensorscope Project

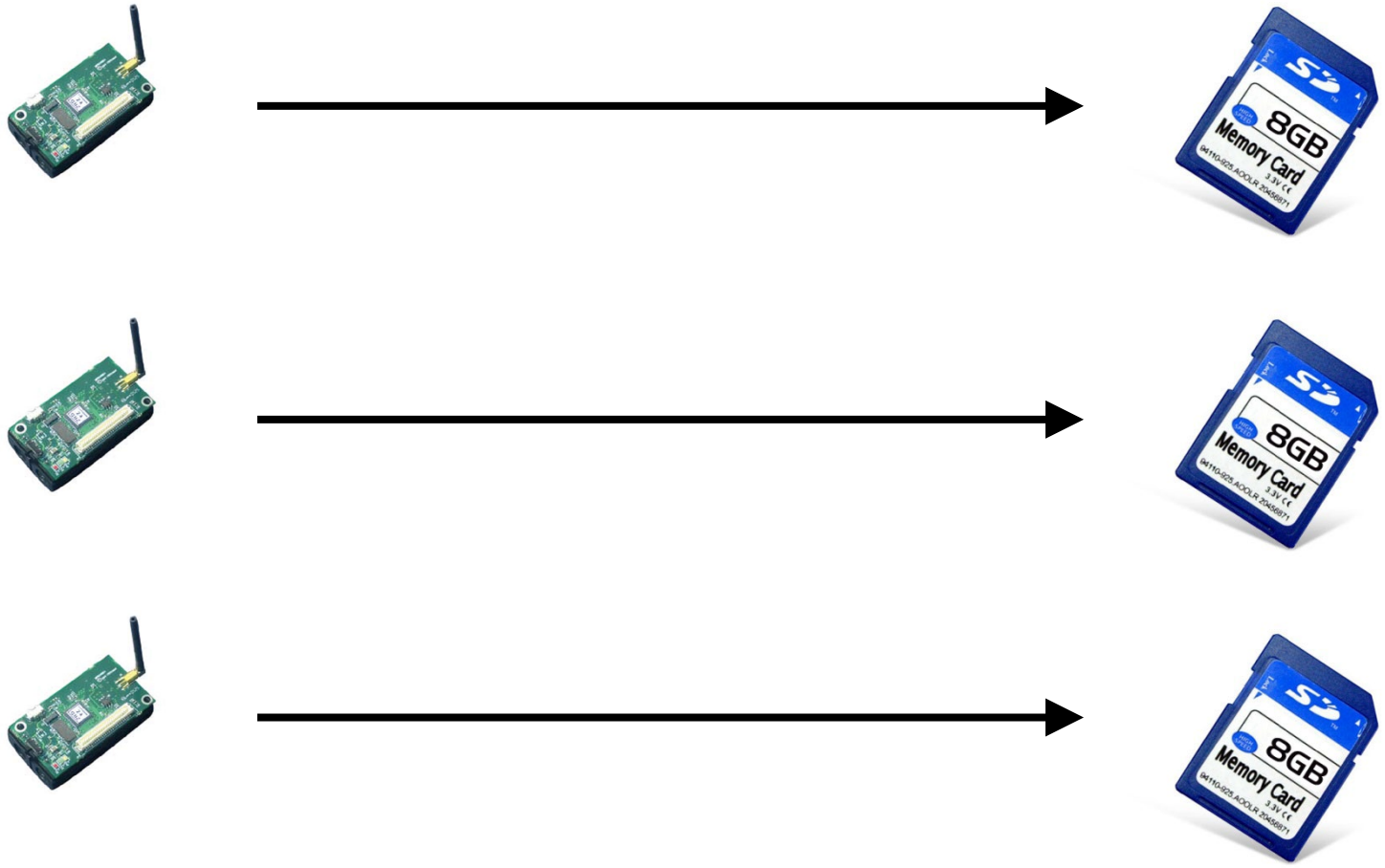
# Introduction



# Topology

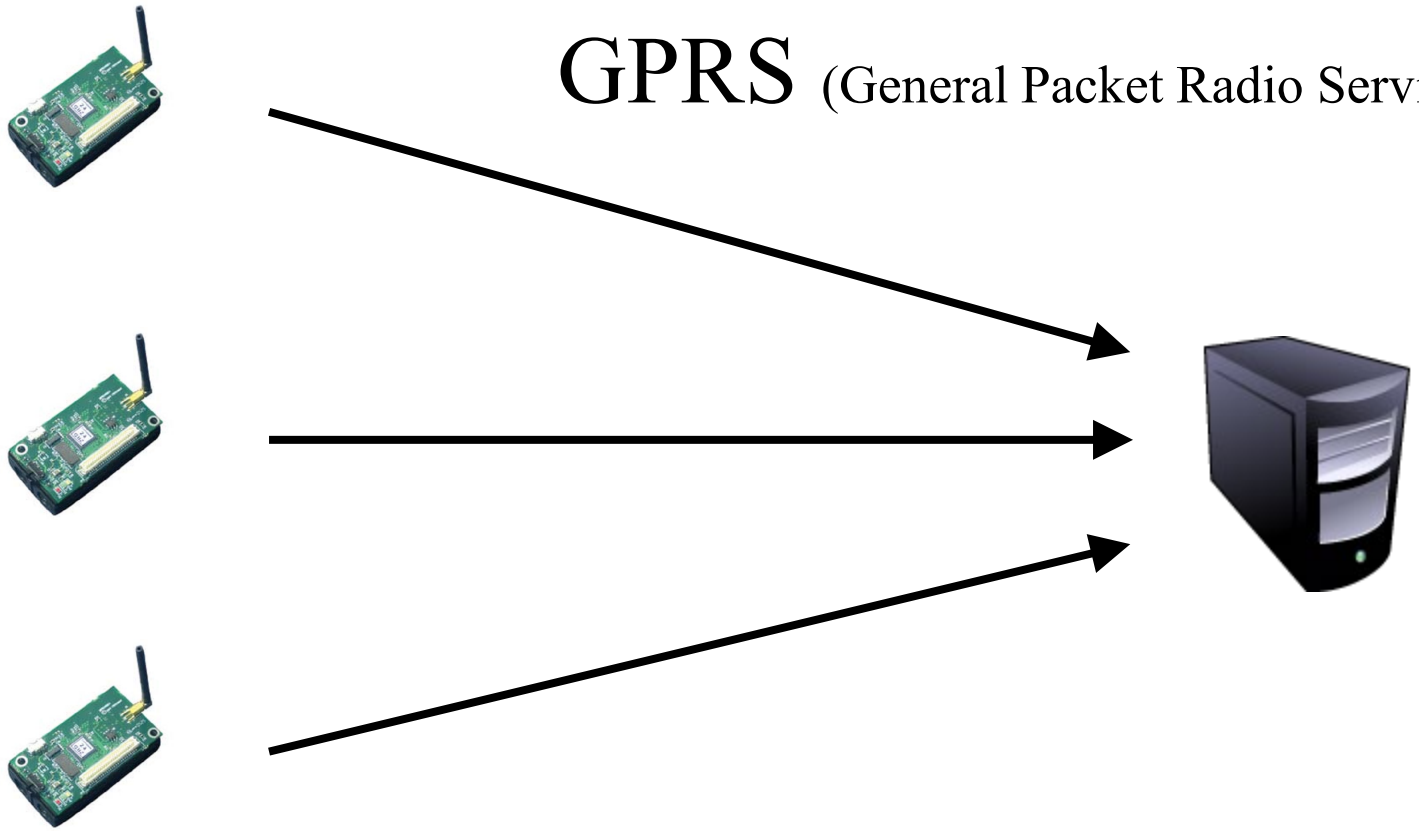


# Topology



# Topology

# Topology



# Topology

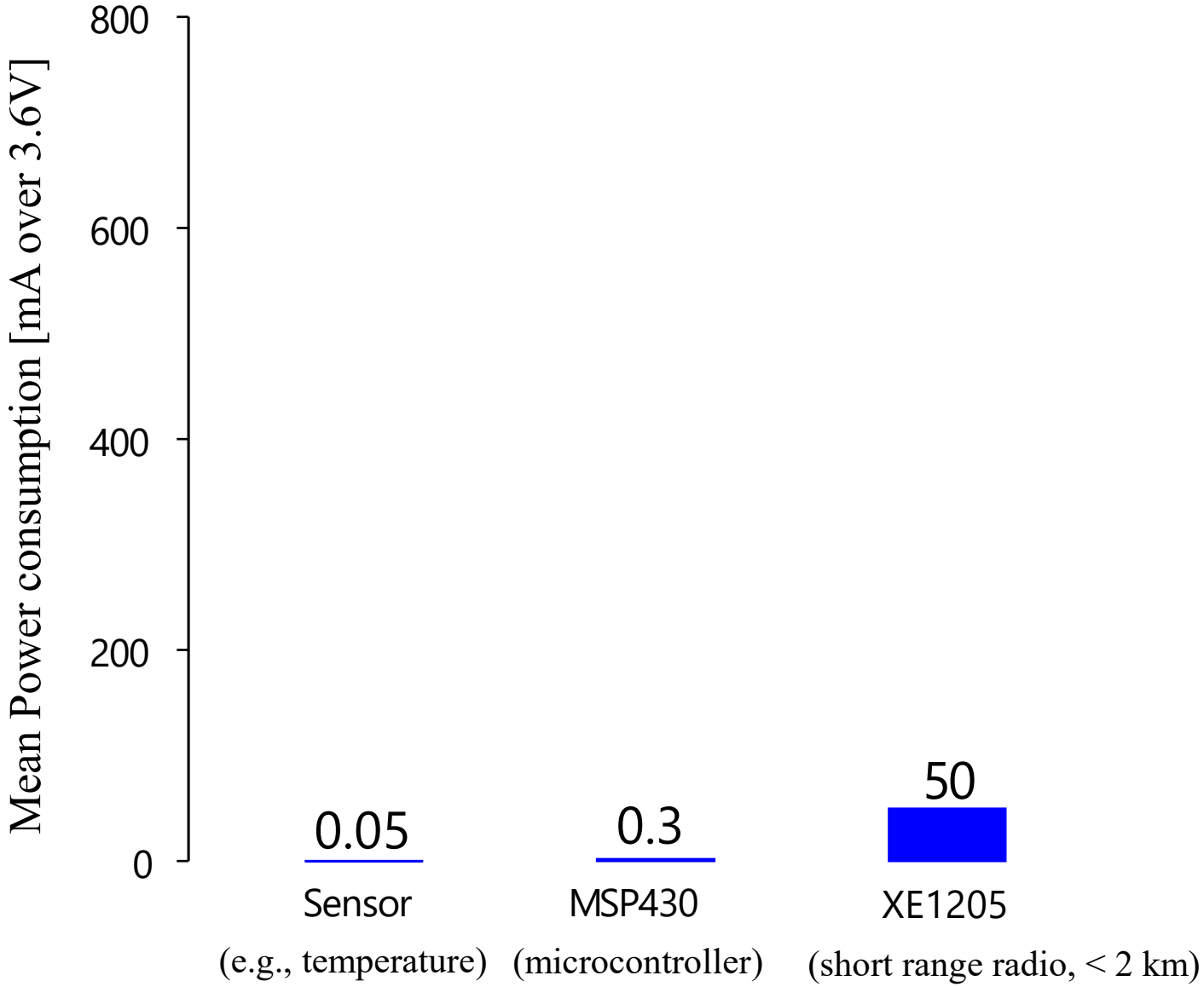
**Pros**

- Very simple!
- Essentially no restrictions in sensor locations

**Cons**

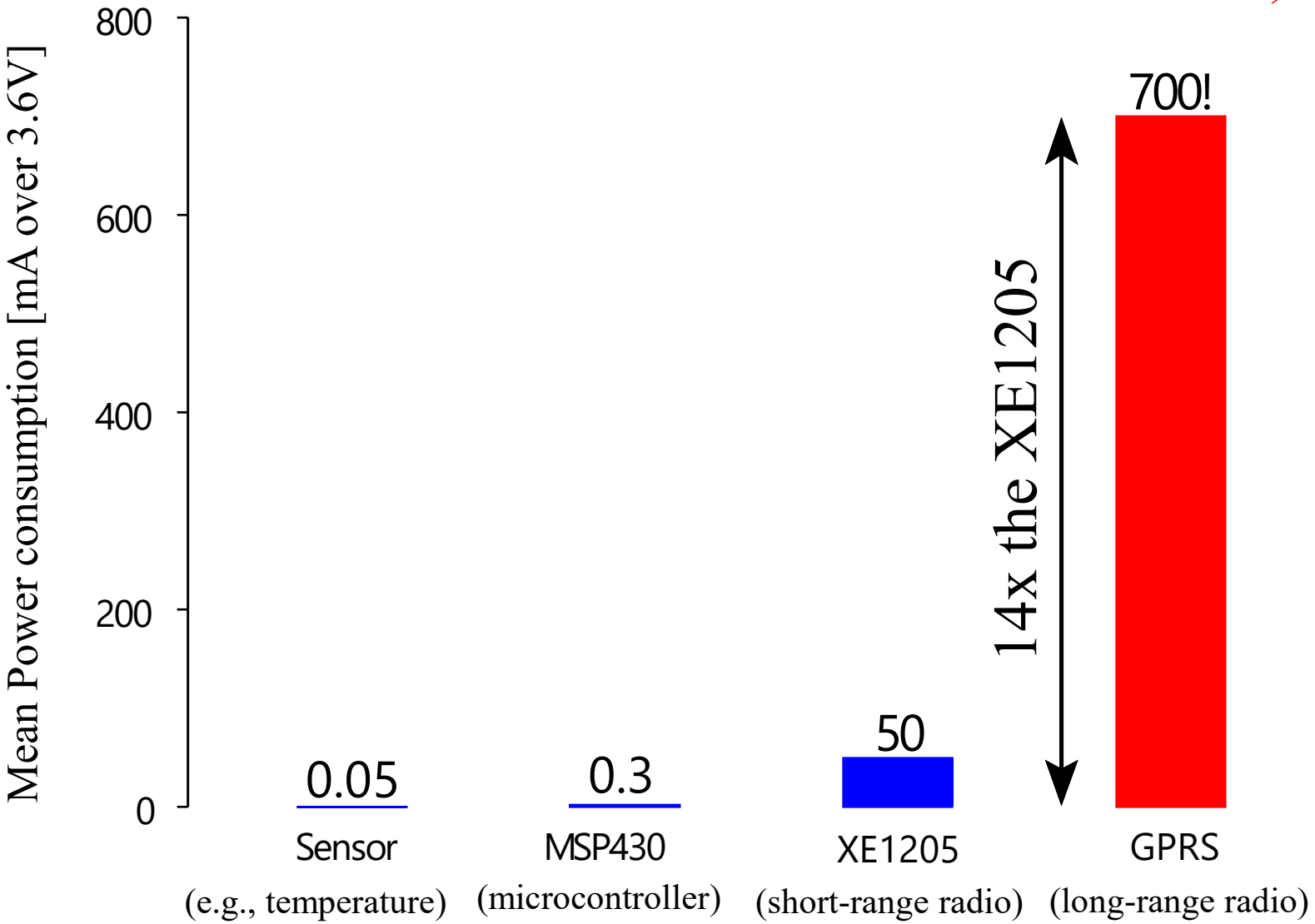
- The closest server access point may be quite far from the stations
- A long-range link may consume a lot of energy!

# Topology


Mean Power consumption [mA over 3.6V]
800
600
400
200
0
0.05
0.3
50
Sensor
MSP430
XE1205
(e.g., temperature)
(microcontroller)
(short range radio, < 2 km)

# Power Consumption

See s. 55, W6

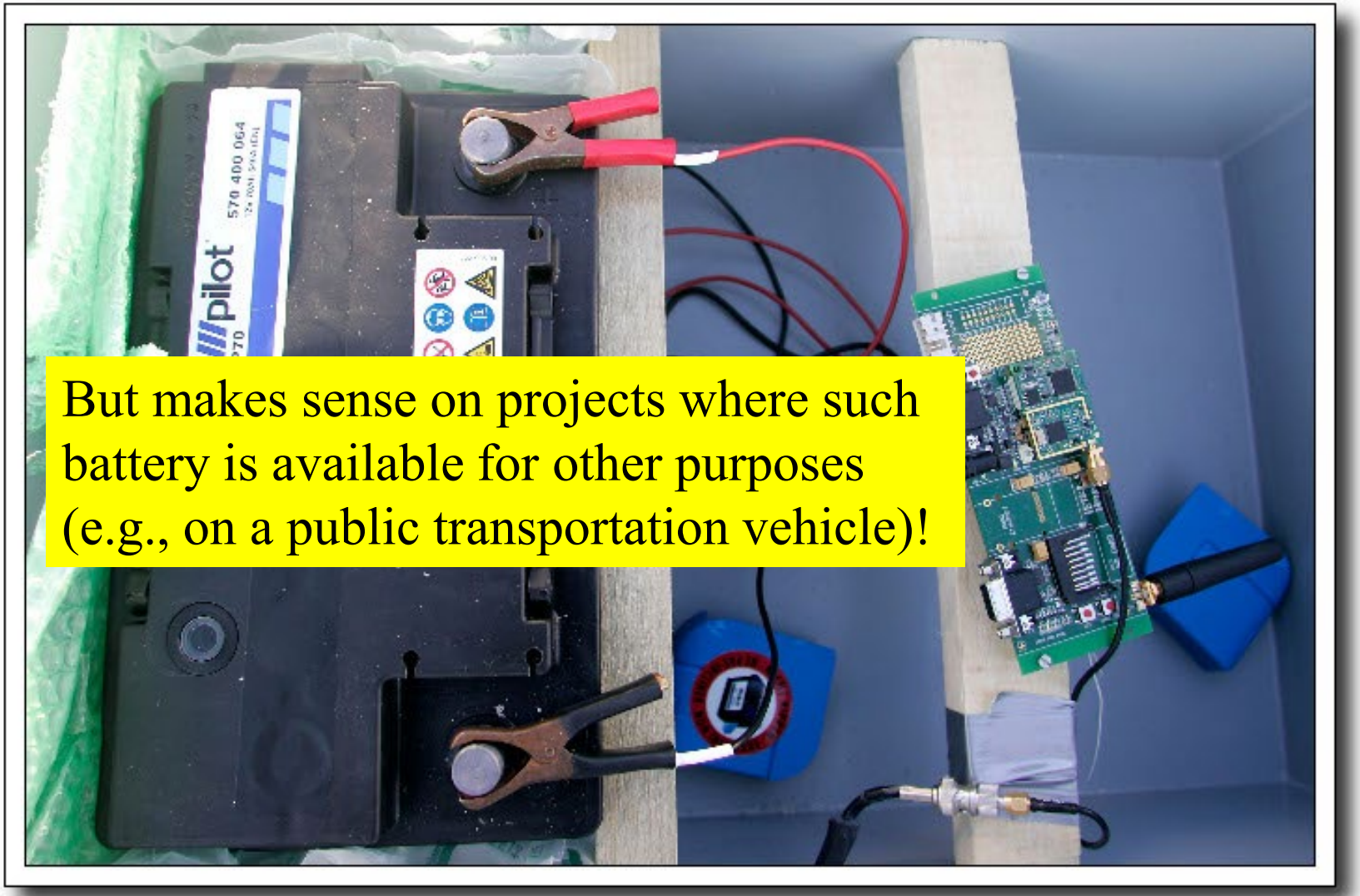


# Power Consumption – Battery Depletion Without Charging

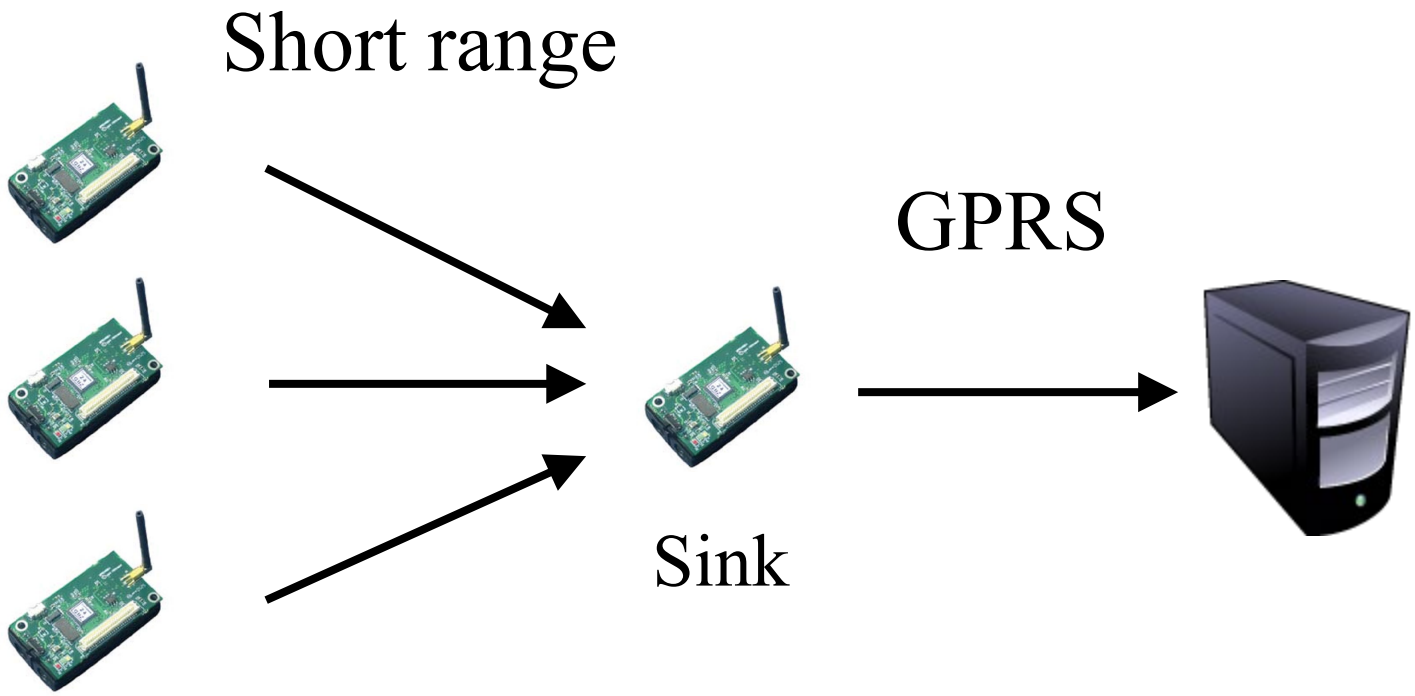
See s. 56, W6

- With NiMH batteries (nominal voltage 1.2 V), 12 Ah, total energy 14.4 Wh, and only one of the components below always active, the battery will be depleted in:
  - Sensor: **9 years**
  - MSP430: **1.5 years**
  - Short-range radio: **3 days**
  - Long-range radio: **6 hours**

# Topology


pilot
570 400 064
But makes sense on projects where such battery is available for other purposes (e.g., on a public transportation vehicle)!

# Topology



# Topology

Recall Friis law again:

See s. 38, W7

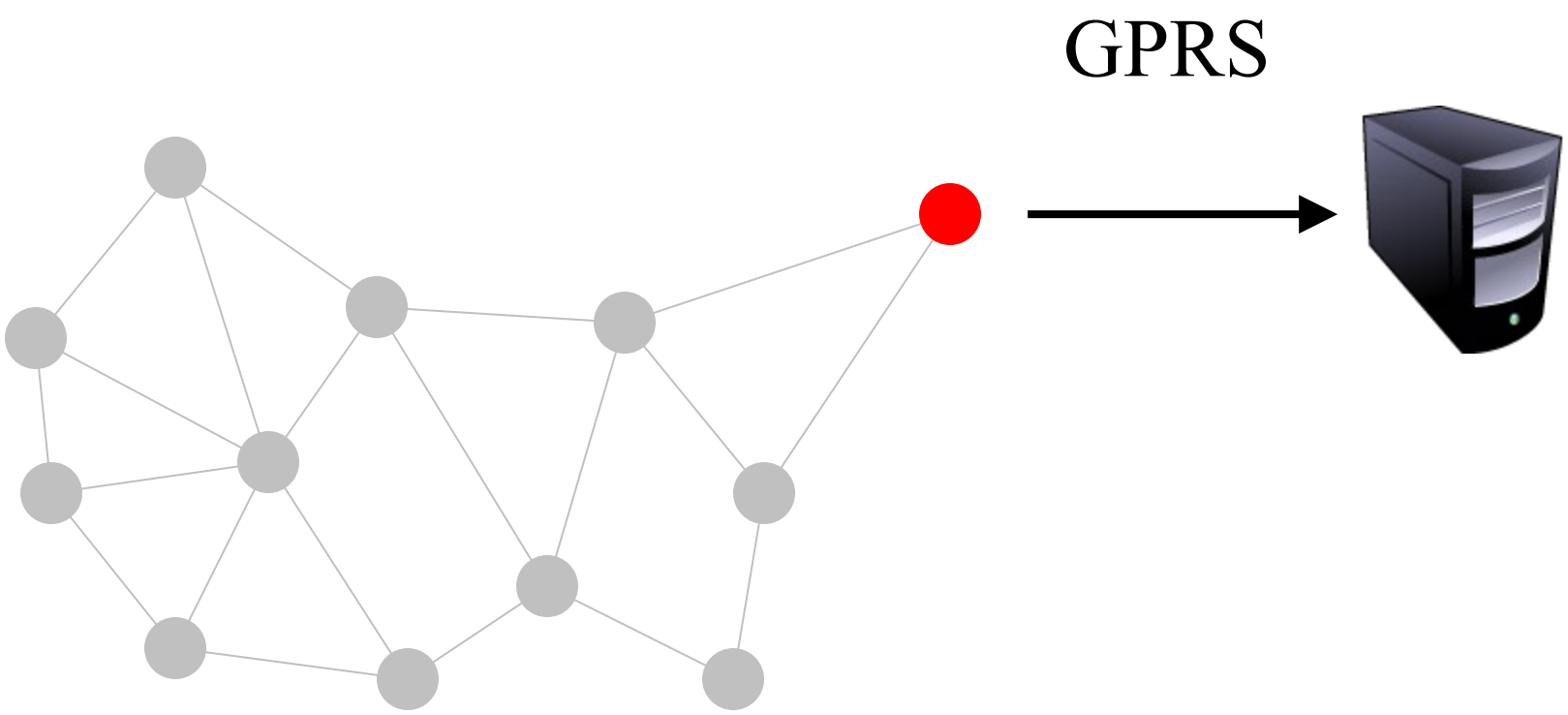
$$L=\left(\frac{4\pi df}{c}\right)^2$$

Example: To transmit over 5 Km, using a 868 MHz carrier, we can:

- One hop of 5 km: **L = 106 dB**
- Two hops of 2.5 km: **L = 99 dB**
- Five hops of 1 km: **L = 92 dB**

Energy is the main issue !!!

# Multi-hop Sensor Network



# Multi-hop Sensor Network

**Pros**

- Only one car battery in the network
- The sensor network has extended monitoring coverage
- Multiple routes for stations to communicate with the sink
- Auto configurable network (robustness)

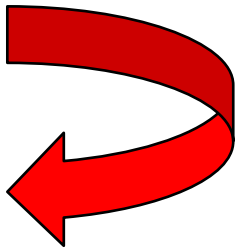
**Cons**

- Significantly more complicated
- Data rate is not increased
- Unable to use directional antennas

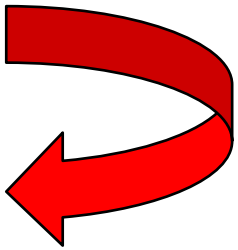
# Multi-hop WSNs

Implementation:

- **Neighborhood discovery**
- Data routing
- **Time synchronization**
- Duty-cycling (radio management)



Pre-condition

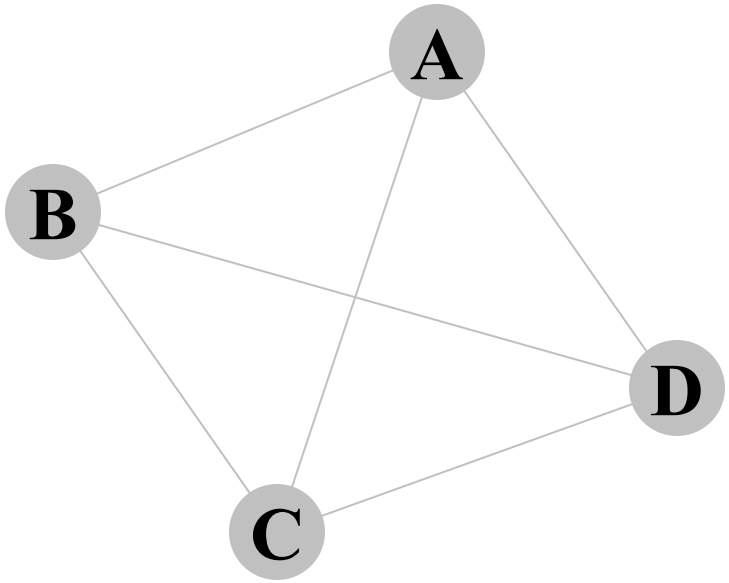


Pre-condition

# Neighborhood

Hello messages (Beacons) are one common method:

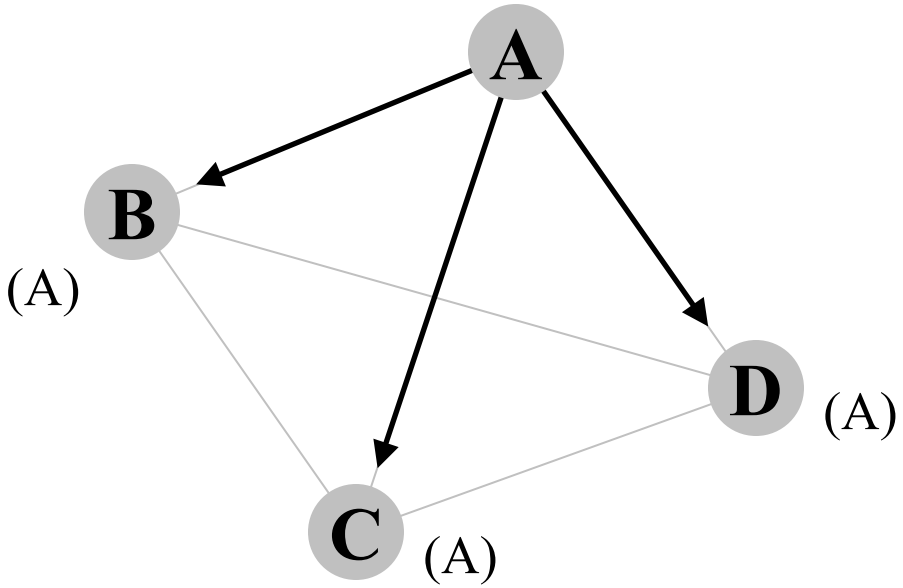
1. Node A sends a HELLO message to its neighbors (B, C, and D).
2. Nodes B, C, and D update their neighborhood table.
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …



# Neighborhood

Hello messages (Beacons) are one common method:

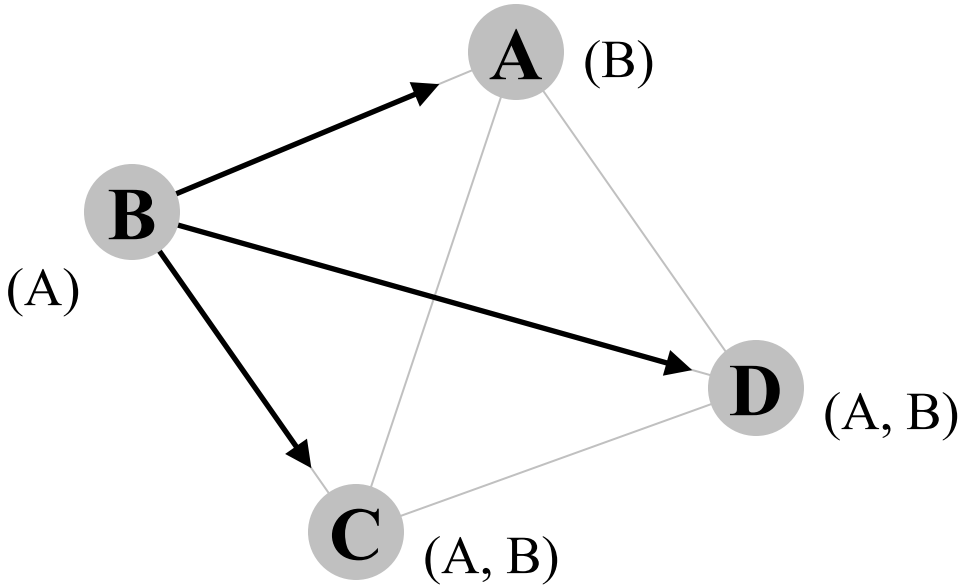
1. Node A sends a HELLO message to its neighbors (B, C, and D).
2. Nodes B, C, and D update their neighborhood table.
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …



# Neighborhood

Hello messages (Beacons) are one common method:

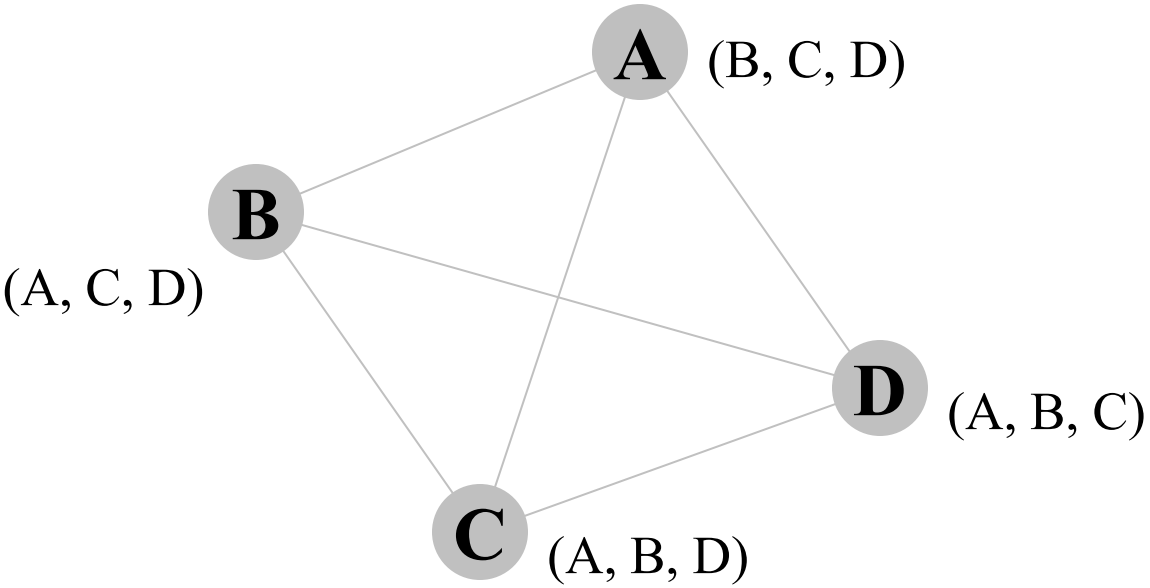
1. Node A sends a HELLO message to its neighbors (B, C, and D).
2. Nodes B, C, and D update their neighborhood table.
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …



# Neighborhood

Hello messages (Beacons) are one common method:

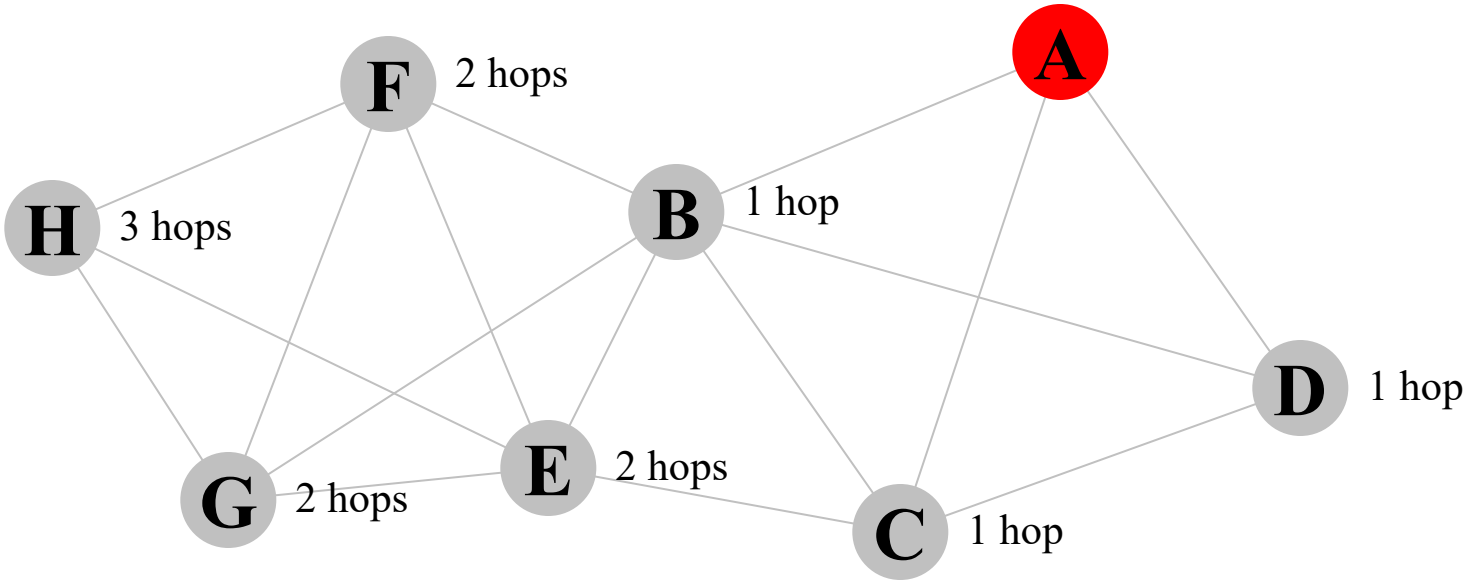
1. Node A sends a HELLO message to its neighbors (B, C, and D).
2. Nodes B, C, and D update their neighborhood table.
3. Node B sends a HELLO message to its neighbors (A, C, and D).
4. …



# Neighborhood

What information do we need about our neighbors?

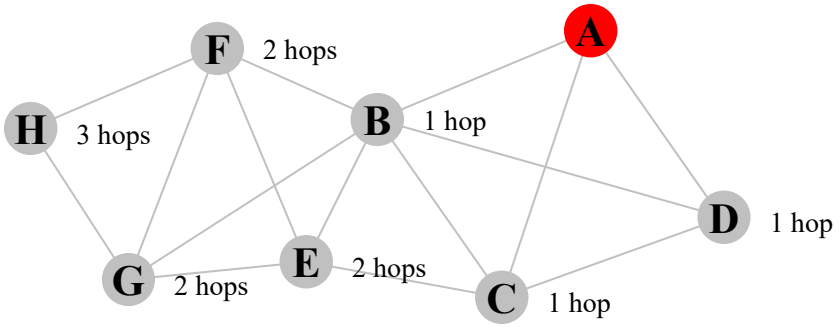
- Distance to sink
- Last time heard
- Link quality



# Neighborhood

Node E's neighborhood table

| Id | Age | Distance | Quality |
|---|---|---|---|
| B | 2 min | 1 hop | 87% |
| C | 2 min | 1 hop | 98% |
| F | 4 min | 2 hops | 74% |
| G | 1 min | 2 hops | 93% |




A few remarks:

- Only the distance to the sink is stored in # of hops
- Neighborhood discovery can't be done only once!
- We need to estimate link qualities!

# Neighborhood

Variations of simple schema:

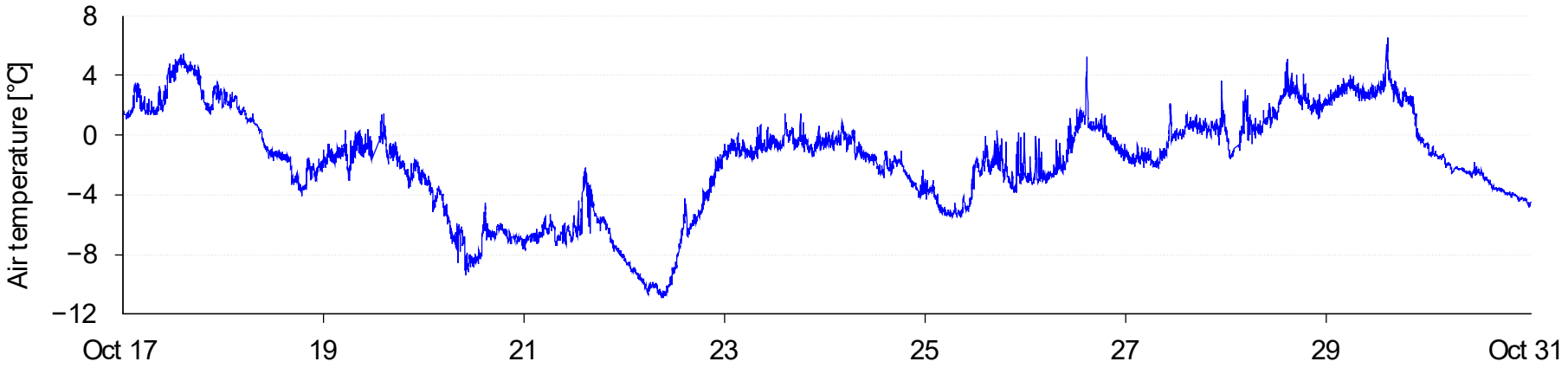
- Each node sends X beacons per minute.
- Number of beacons received per minute are stored.
- Quality is estimated over the past Y minutes by counting losses.

Example (X = 10;Y = 4):

| t – 4 | t – 3 | t - 2 | t - 1 | |
|---|---|---|---|---|
| 10 | 7 | 8 | 8 | Quality = 0.8 |
| 7 | 8 | 8 | 6 | Quality = 0.71 |
| 8 | 8 | 6 | 5 | Quality = 0.64 |

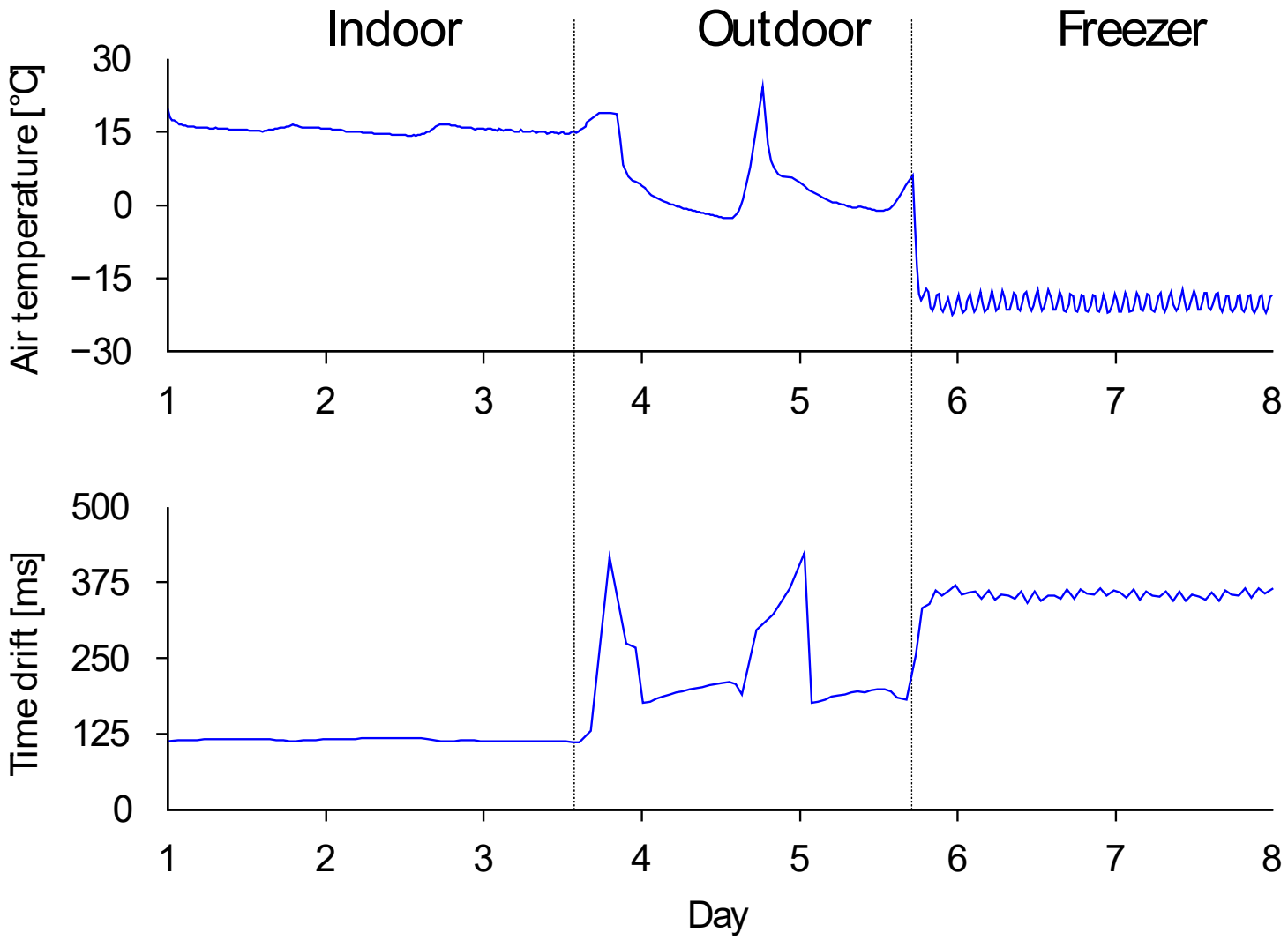
# Time Synchronization

Weather conditions, especially temperature and humidity, may have a significant effect on hardware




Crystal oscillators are highly impacted by temperature!

# Time Synchronization



# Time Synchronization

Nodes need to know the time to:

- Timestamp packets
- Synchronize actions (e.g., taking samples, transmitting data)

How do we get time:

- Fully decentralized: Every node gets the time itself
- Partially centralized: Time is propagated from reference nodes

# Time Synchronization

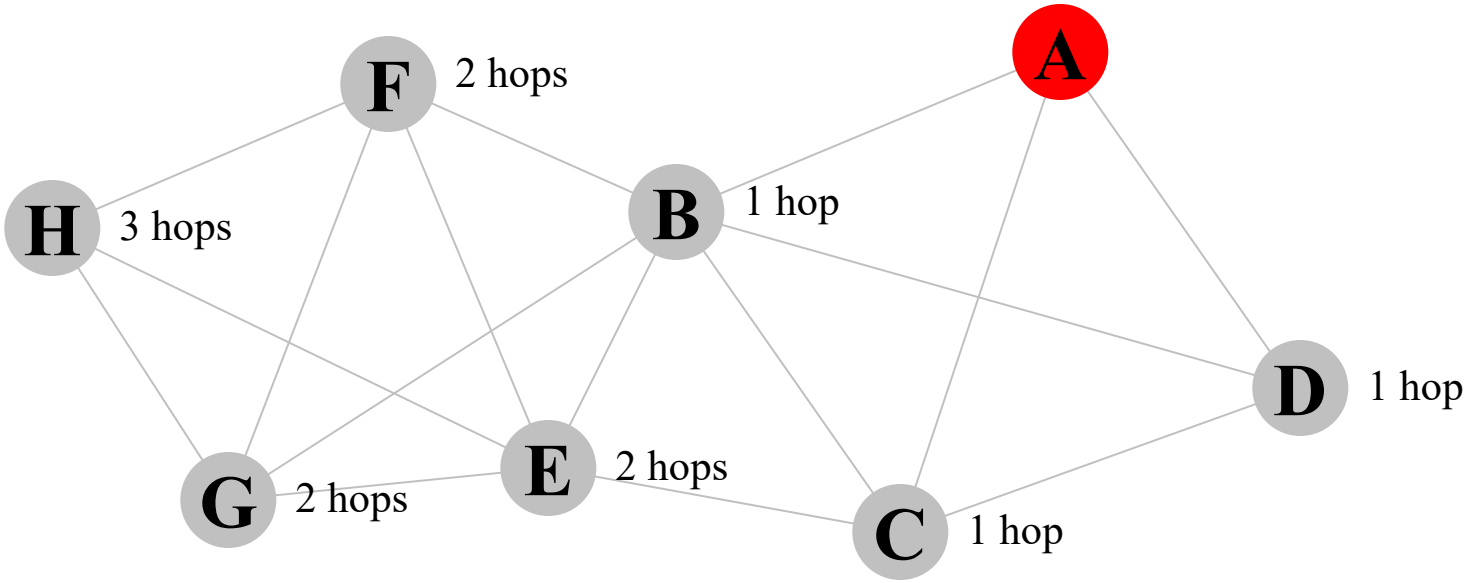
Every node gets the time:

- Atomic clock receivers:
  - Cheap (both energy and $)
  - Complexity
  - Limited coverage
- GNSS:
  - Coverage
  - Complexity
  - High cost (energy and $)
- GPRS: same as GNSS with less coverage

**What about a partially centralized approach?**

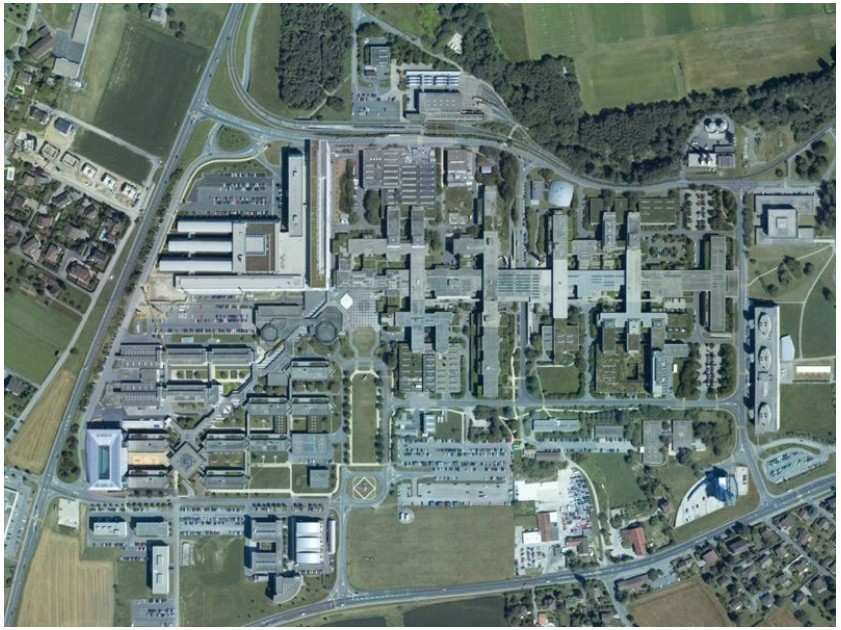
# Time Synchronization

For instance, the sink serve as time reference node leveraging the anyway needed GPRS connection



# SensorScope

Many previous successful deployments

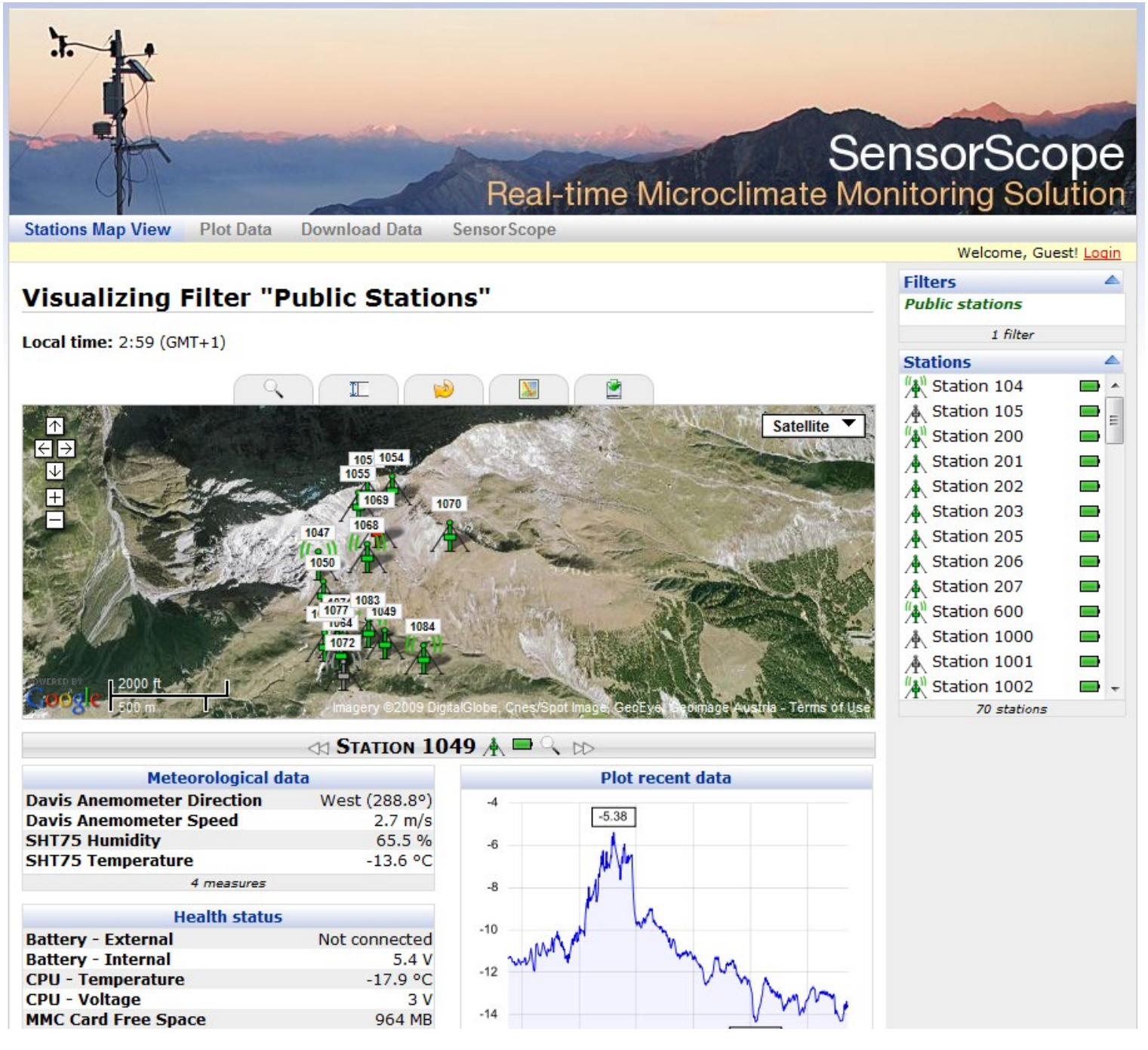


97 stations deployed at EPFL (one year)

# SensorScope

Many previous successful deployments

16 stations deployed at Le Génépi to monitor conditions leading to dangerous mudslides (two months)


SensorScope
Real-time Microclimate Monitoring Solution
Stations Map View
Plot Data
Download Data
SensorScope
Welcome, Guest! Login
Visualizing Filter "Public Stations"
Local time: 2:59 (GMT+1)
Satellite
105
1054
1055
1069
1070
1047
1068
1050
1083
1077
1049
1084
1072
2000 ft
500 m
Imagery ©2009 DigitalGlobe, Cnes/Spot Image, GeoEye, Geoimage Austria - Terms of Use
Filters
Public stations
1 filter
Stations
Station 104
Station 105
Station 200
Station 201
Station 202
Station 203
Station 205
Station 206
Station 207
Station 600
Station 1000
Station 1001
Station 1002
70 stations
Station 1049
Meteorological data
Davis Anemometer Direction West (288.8°)
Davis Anemometer Speed 2.7 m/s
SHT75 Humidity 65.5 %
SHT75 Temperature -13.6 °C
4 measures
Health status
Battery - External Not connected
Battery - Internal 5.4 V
CPU - Temperature -17.9 °C
CPU - Voltage 3 V
MMC Card Free Space 964 MB
Plot recent data
-4
-6
-8
-10
-12
-14
-5.38

# Conclusion

# Take Home Messages

- Nonlinear motion and measurement models can be linearized and their output fused with an Extended Kalman Filter to solve 2D (and 3D) localization problems with real vehicles (e.g., differential-drive wheeled robots)
- Leveraging exteroceptive sensing data for localization in practice means being able to extract information about features in a way that is comparable to known environmental mapping data
- Sensor networks are a concrete response to monitor environments which are way larger than the field of view of single sensing assets
- The hardware technology of a sensor network is represented by that available at the single node level but the software stack must be designed for distributed operation
- Intelligent instruments are very powerful and characterized by an increased software complexity which offer new opportunities in terms of customization, automation, etc.
- Real-world deployments may be highly unpredictable: design robustness of software and hardware is key!

# Additional Literature – Week 11

**Books**

- Siegwart R., Nourbakhsh I., and Scaramuzza D., "Introduction to Autonomous Mobile Robots, second Edition", MIT Press, 2011.
- Thrun S., Burgard W., and Fox D., Probabilistic Robotics, MIT Press, 2005.

**Pointers**

- Permasense: https://www.permasense.ch
- GITEWS: https://www.gitews.de

**Papers**

- Culler D., Estrin D., and Srivastava M., "Guest Editors' Introduction: Overview of Sensor Networks". *IEEE Computer*, Vol. 37, No. 8, pp.41-49, 2004.
- Corke P., Wark T., Jurdak R., Hu W., Valencia P., Moore D., "Environmental Wireless Sensor Networks", *IEEE Proceedings*, Vol. 98, No. 11, pp. 1903-1917, 2010.