Localizing Sensors from their Responses to Targets

Shigeo Shioda

Graduate School of Engineering
Chiba University, Japan

E-mail: shioda@faculty.chiba-u.jp
http://www.qos.tu.chiba-u.jp

❶ Introduction

❷ Classification of Localization Methods

❸ Distance-Matrix-Based Localization

❹ Numerical Example

❺ Localization via Responses to Targets

❻ Numerical Example

❼ Summary & Future Work

Sensor Localization

What?

- Technique estimating locations of sensors.
- It can be used to estimate locations of persons (objects) carrying sensors.

Why?

- Data collected by a sensor is meaningful in conjunction with knowledge of sensor locations.
- Sensor locations are valuable for management & control of sensor networks (for example, routing).
- Localization is a mathematically interesting subject.

This talk mainly focuses on theoretical aspects of localization.

Classification of Localization Methods

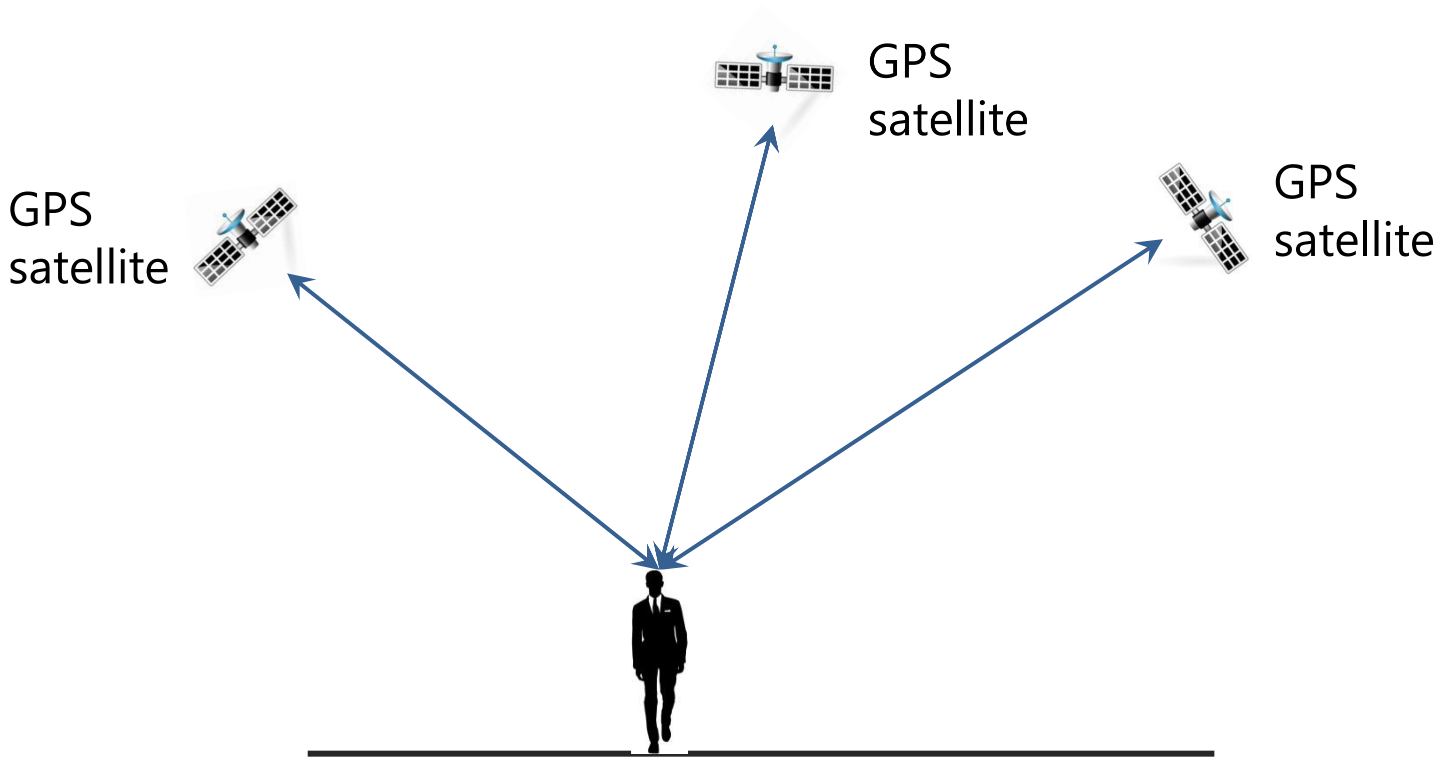
- Range-Based Method

- Range-Free (Connectivity-Based) Method

Range-Based Localization

Range-Based Localization

- Localization based on the distances to the reference points (anchors)
- Extra hardware for distance measurement is required



Range-Free Localization

Connectivity

- Proximity relationship between sensors.
- For example, sensors A and B being connected means that A and B are located in close proximity.
- Capability of direct wireless communication

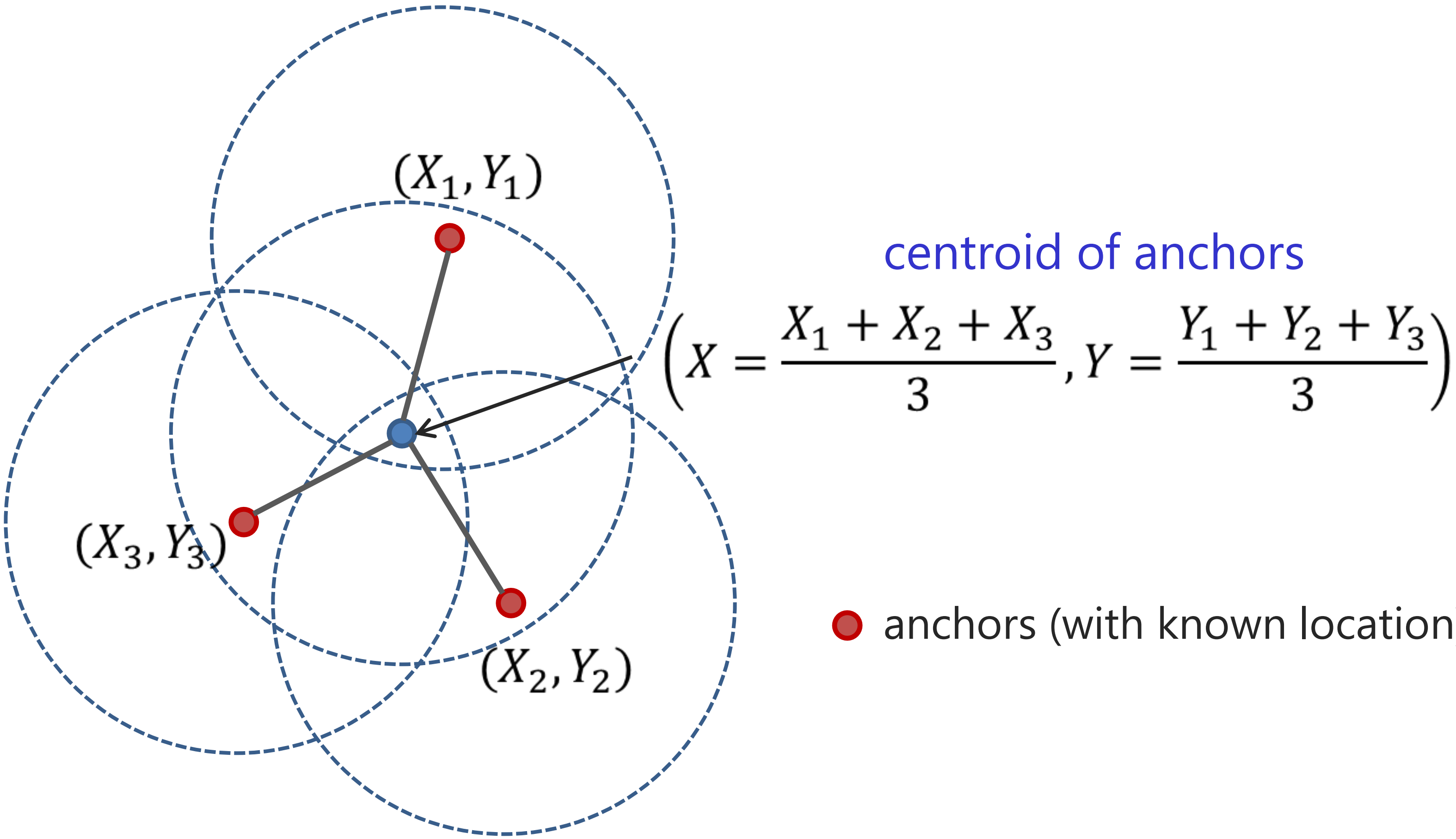
Range-Free (Connectivity-Based) Localization

- Localization based on connectivity to reference points
- Less accurate than range-based localization
- Implementation is easier.

Range-Free Localization

Example: Centroid Algorithm

- Estimates the location as the centroid of anchors in the neighborhood.



$$\left(X = \frac{X_1 + X_2 + X_3}{3}, Y = \frac{Y_1 + Y_2 + Y_3}{3}\right)$$

1. Introduction
2. Classification of Localization Methods
3. **Distance-Matrix-Based Localization**
4. Numerical Example
5. Localization via Responses to Targets
6. Numerical Example
7. Summary & Future Work

Distance-Matrix-Based Localization

Distance Matrix

- Contains (measured) pairwise distances between sensors.
- May have missing elements (when some pairwise distances cannot be measured).
- Real distance (range base) or hop count (range free)

Distance-Matrix (DM) Based Localization

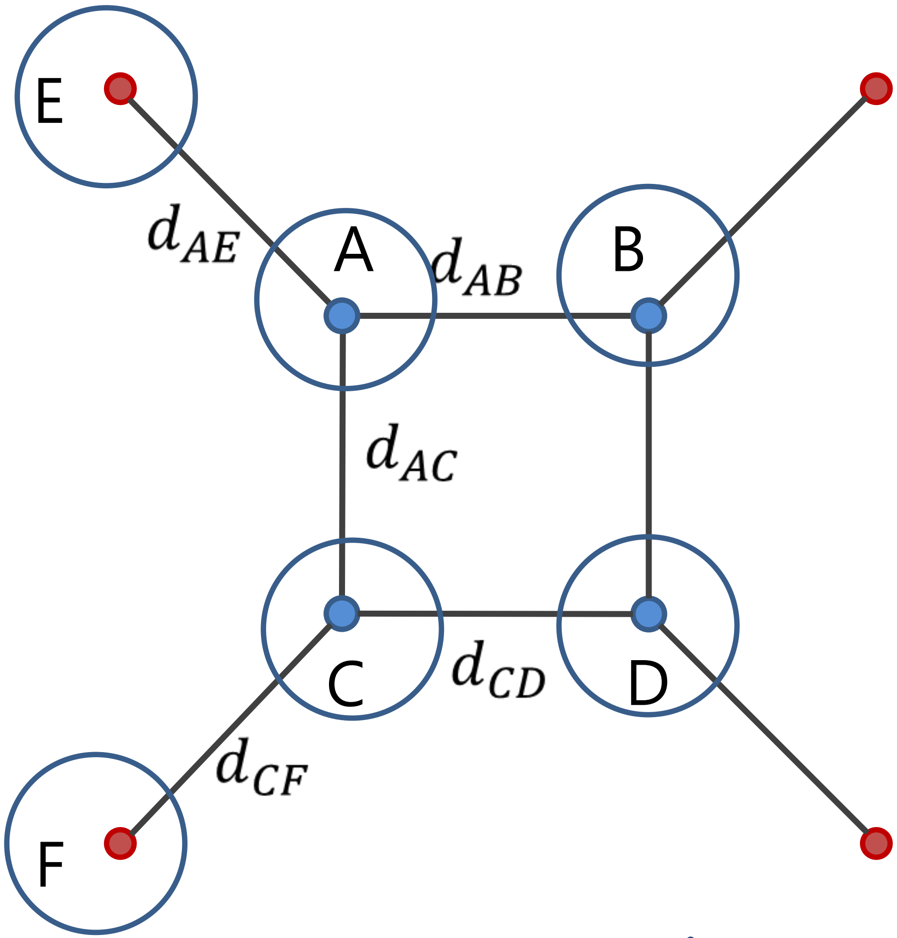
- Localization based on the DM
- Requires only a few anchors

  (Sensors act as anchors for their neighbor sensors to each other.)

Distance-Matrix-Based Localization

Example

- anchors (known location)
- sensors (unknown location)



Sensor A is localized based on
- distances d_{AB}, d_{AC}, d_{AE}
- location estimates of B and C
- location of anchor E

Sensor C is localized based on
- distances d_{AC}, d_{CD}, d_{CF}
- location estimates of A and D
- location of anchor F

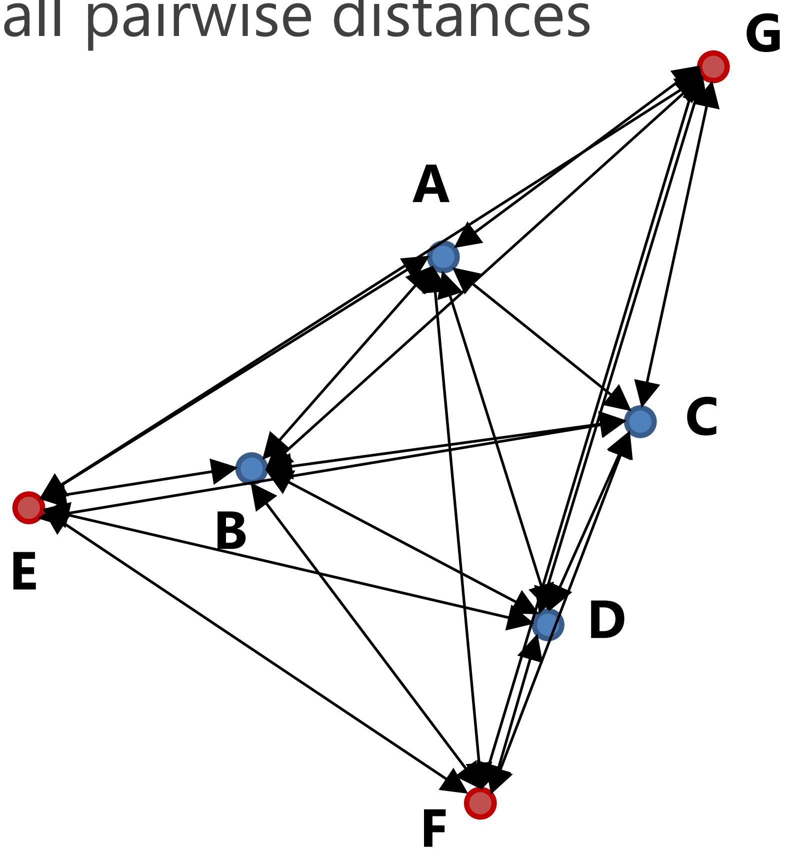
Cooperative Localization

Distance Matrix

■ (Range Based) Distance Matrix

– Contains measured real pairwise distances
– No missing element when all pairwise distances can be measured.

	A	B	C	D	E	F	G
A		1.2	1.3	1.5	2.3	2.5	1.4
B	1.2		1.5	1.4	0.9	2.1	2.6
C	1.3	1.5		0.9	2.6	2.1	1.7
D	1.5	1.4	0.9		2.1	0.8	2.5
E	2.3	0.9	2.6	2.1		1.6	3.3
F	2.5	2.1	2.1	0.8	1.6		3.1
G	1.4	2.6	1.7	2.5	3.3	3.1	

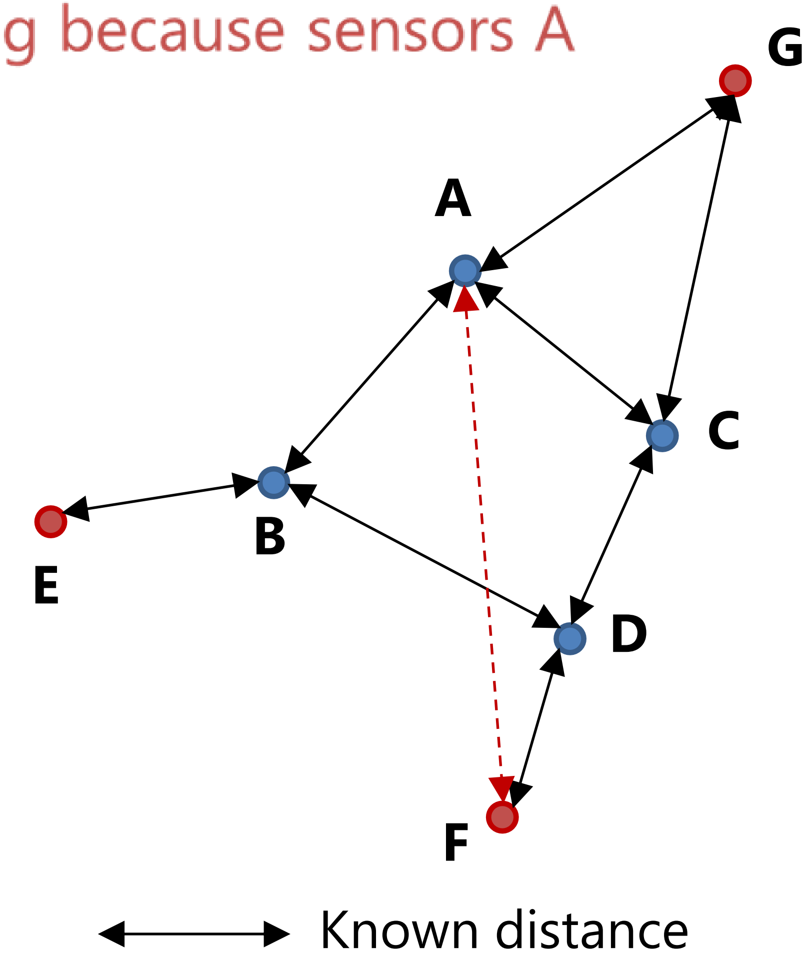


Distance Matrix

(Range Based) Distance Matrix

- Usually, missing elements exist.
- For example, d_{AF} is missing because sensors A

	A	B	C	D	E	F	G
A		1.2	1.3				1.4
B	1.2			1.4	0.9		
C	1.3			0.9			1.7
D		1.4	0.9			0.8	
E		0.9					
F				0.8			
G	1.4		1.7				



Distance Matrix

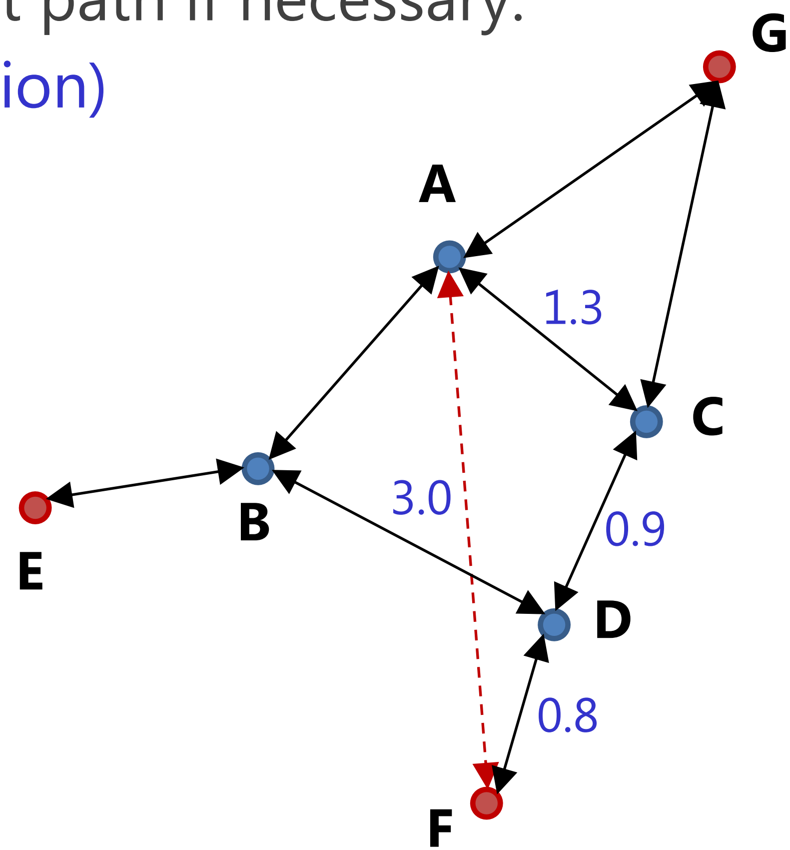
■ (Range Based) Distance Matrix

– Missing element can be compensated by the distance along the shortest path if necessary.

(Shortest Path Approximation)

	A	B	C	D	E	F	G
A		1.2	1.3	2.2	2.1	3.0	1.4
B	1.2		2.3	1.4	0.9	2.2	2.6
C	1.3	2.3		0.9	3.2	1.7	1.7
D	2.2	1.4	0.9		2.3	0.8	2.6
E	2.1	0.9	3.2	2.3		3.1	3.5
F	3.3	2.2	1.7	0.8	3.1		3.4
G	1.4	2.6	1.7	2.6	3.5	3.4	

Elements-compensated DM

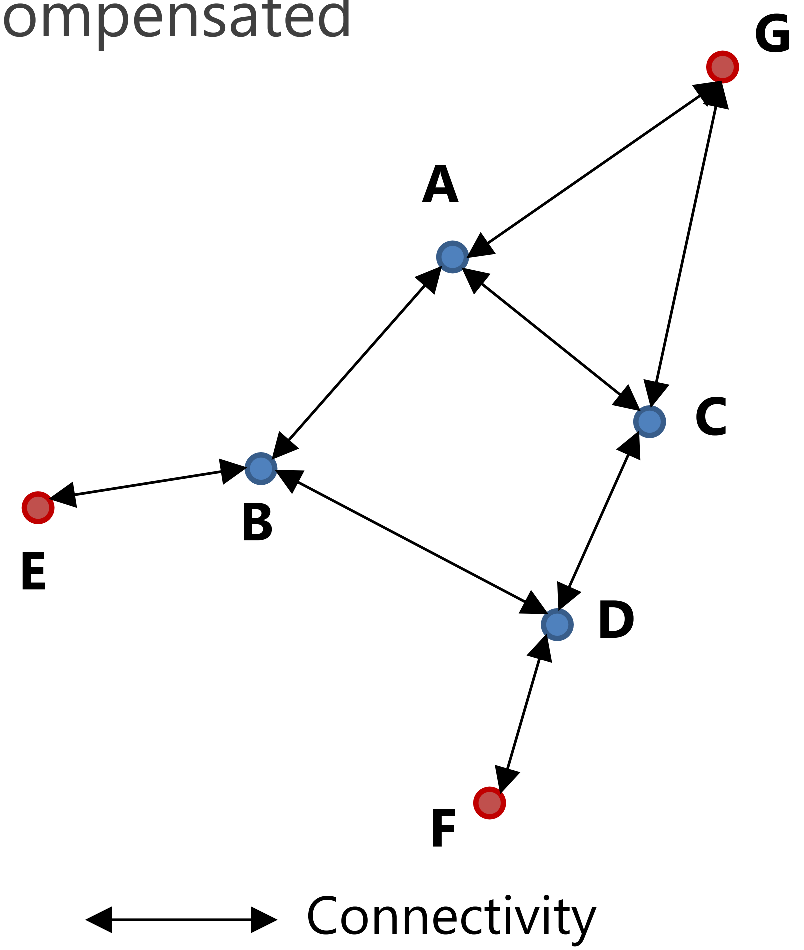


Distance Matrix

(Range Free) Distance Matrix

- Represents connectivity information (Adjacent Matrix).
- Missing elements can be compensated by hop counts.

	A	B	C	D	E	F	G
A		1	1	2	2	3	1
B	1		2	1	1	2	2
C	1	2		1	3	2	1
D	2	1	1		2	1	2
E	2	1	3	2		2	3
F	3	2	2	1	2		3
G	1	2	1	2	3	3	



DM-Based Localization

- DM-based localization can be formulated as non-linear optimization problem

- **MDS (Multi-Dimensional Scaling)**
  - Originally developed for mathematical psychology
  - Seeks a configuration of sensors in a two (or three) dimensional space that best preserves pairwise distances.
  - Best preserving pairwise distances

    = Minimizing a "Stress function"

Stress Function

■ Stress Function

– Sum of squared difference between estimated and measured inter-sensor distances.

Stress Function

$$\epsilon(X) = \sum_{\{i,j\}} \left(|\vec{x}_i - \vec{x}_j| - d_{ij}^{(m)} \right)^2$$

Summation is taken for sensor pairs whose inter-distance can be measured.

Measured inter-sensor distance $d_{ij}^{(m)}$

Estimated inter-sensor distance $|\vec{x}_i - \vec{x}_j|$

Estimated sensor locations $X = (\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N)^T$

Stress Function

- **If DM has missing elements**
  - Sensor locations are obtained by minimizing stress function using a kind of (steepest) descent method (called stress majorization).

Stress function is not convex in general.

Multi-Dimensional Scaling

■ Classical MDS

- First, compensates missing elements of DM by shortest-path approximation
- Then, applies eigen-decomposition of $X\,X^T$ to obtain sensor locations

■ Metric MDS

- First, assumes some initial location estimates.
- Then, from the initial estimates, decreases stress function step by step using stress majorization without elements compensation.

Multi-Dimensional Scaling

■ Classical MDS

- It does not have local minimum problem (it always yields the unique solution).
- It needs element compensation by shortest-path approximation, which causes estimation error.

■ Metric MDS

- Susceptible to local minimum.
- Elements compensation is not necessary.

Combination of Classical MDS and Metric MDS (Two step localization).

Avoiding Local Minimum

Two Step Localization

1. Compensate missing elements of DM by shortest path approximation.

2. Perform localization (by Classical MDS) based on element-compensated DM.

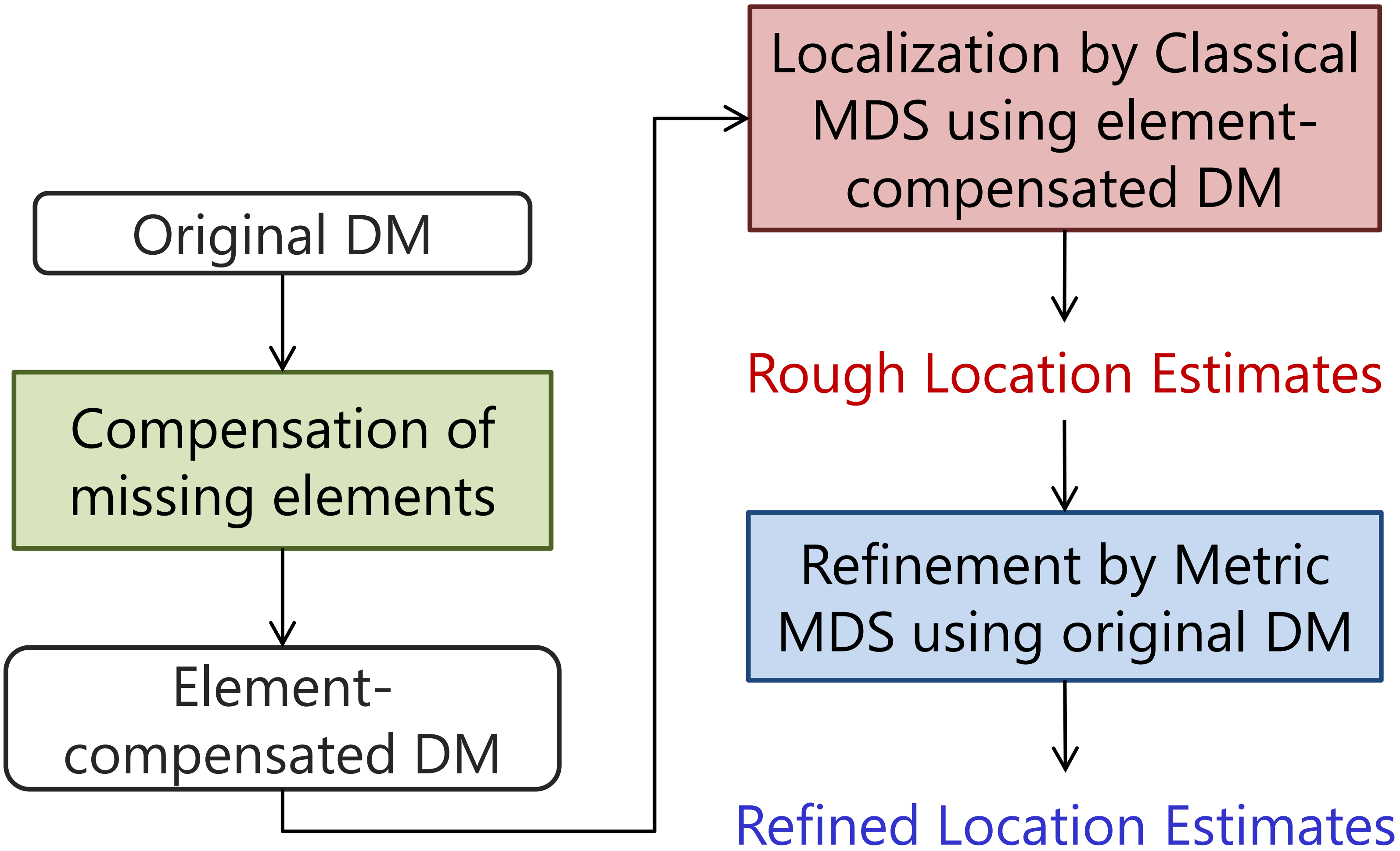
   → Rough Location Estimates (1st step)

3. Make the refinement of the sensor configuration (by Metric MDS) based on the original DM.

   → Refined Location Estimates (2nd step)

Localization Procedure

■ Two Step Localization



Shift, Rotation, and Scaling

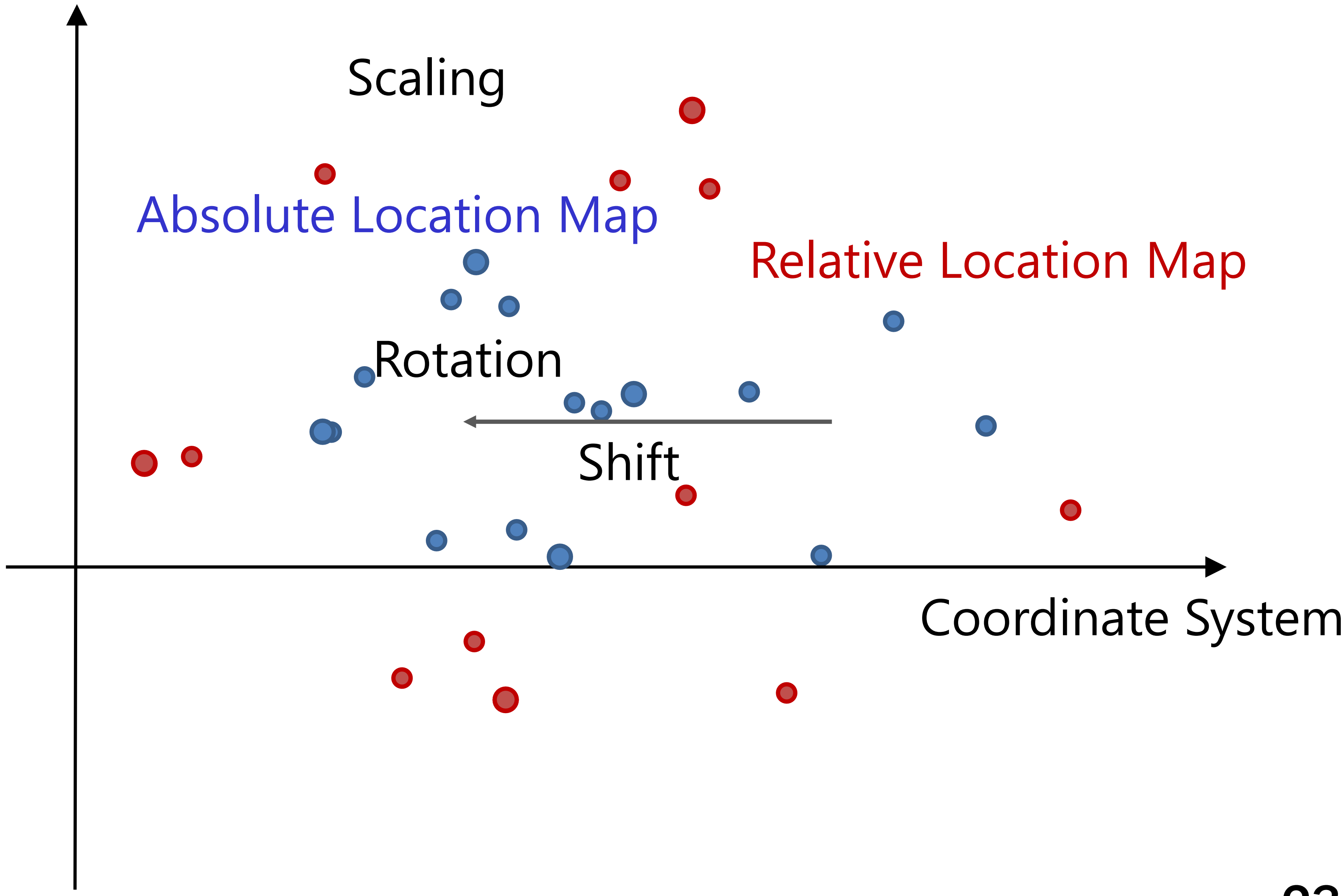
■ Remark

- Range-free localization does not yield the absolute location map, but relative location map.
- It should be transformed into absolute location map by shift, rotation, and scaling.

➢ For the transformation from relative location map to absolute location map, please see

B. Horn, H. Hilden, and S. Negahdaripour, "Closed-from solution of absolute orientation using orthonormal matrices," Journal of the Optical Society of America, vol. 5, no. 7, pp. 1127–1135, 1988.

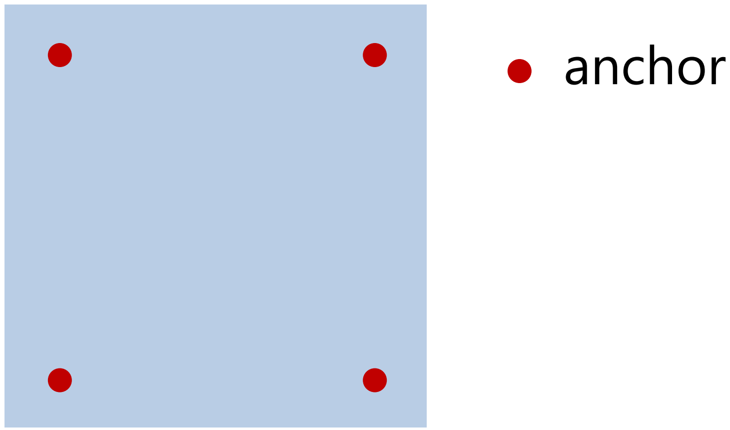
Shift, Rotation, and Scaling



Numerical Example

■ Condition

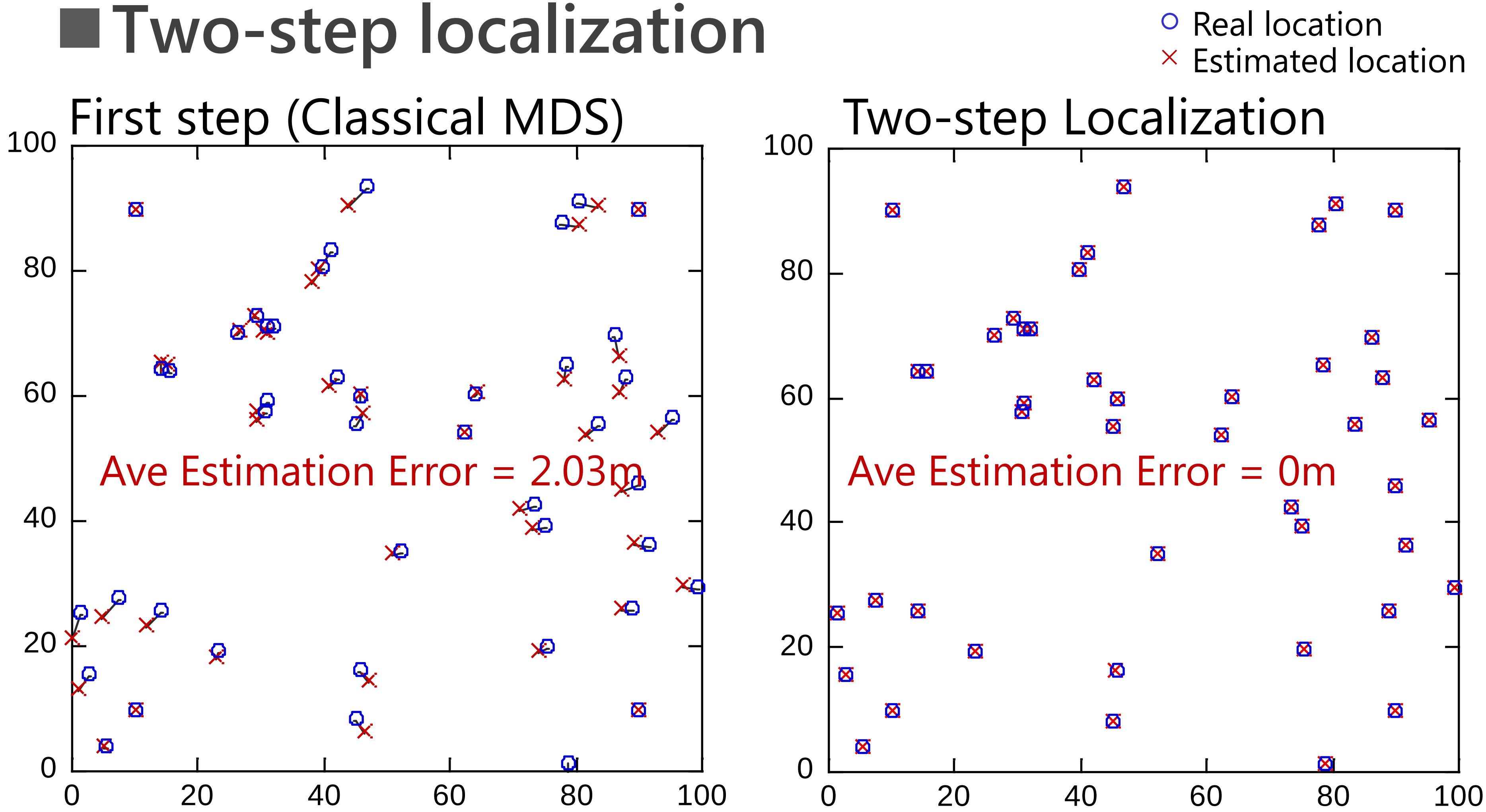
- 40 sensors are randomly deployed in a square region (100 m x 100 m).
- 4 anchors are placed in corners.



- Distance between two sensors cannot be measured if it is greater than 40 m.

Numerical Example

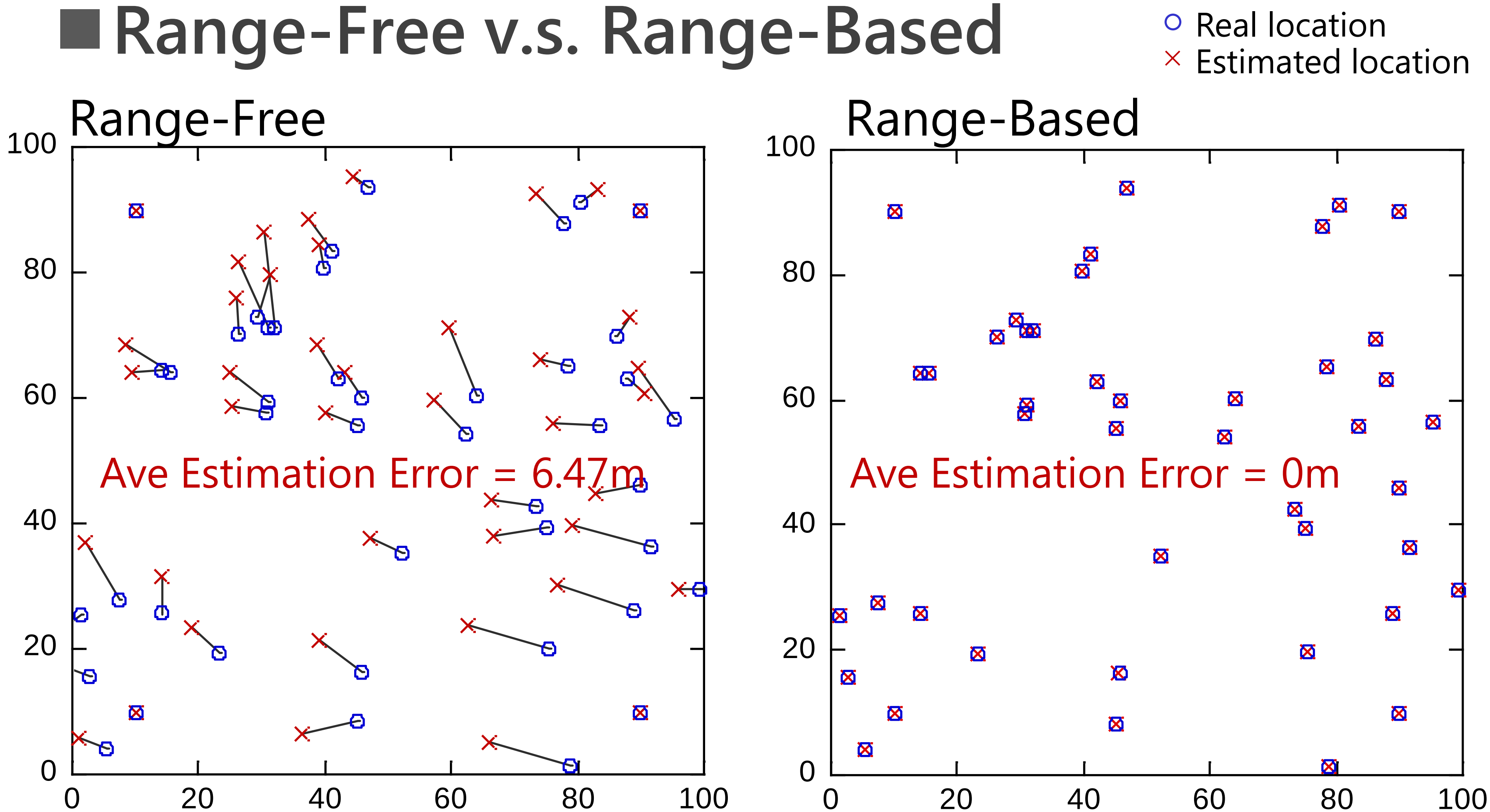
■ Two-step localization



Error in Classical MDS comes from shortest path approximation.

Numerical Example

■ Range-Free v.s. Range-Based



Range-Free is less accurate, but it still yields good estimates.

Numerical Example

■ Distance Measurement Error

If pairwise distance is estimated based on the strength of radio signal (RSS) emitted by each sensor, it should have measurement errors (due to the existence of obstacles)

$$d_{ij}^{(m)} = d_{ij} 10^{-\frac{N_\sigma}{10\eta}}$$

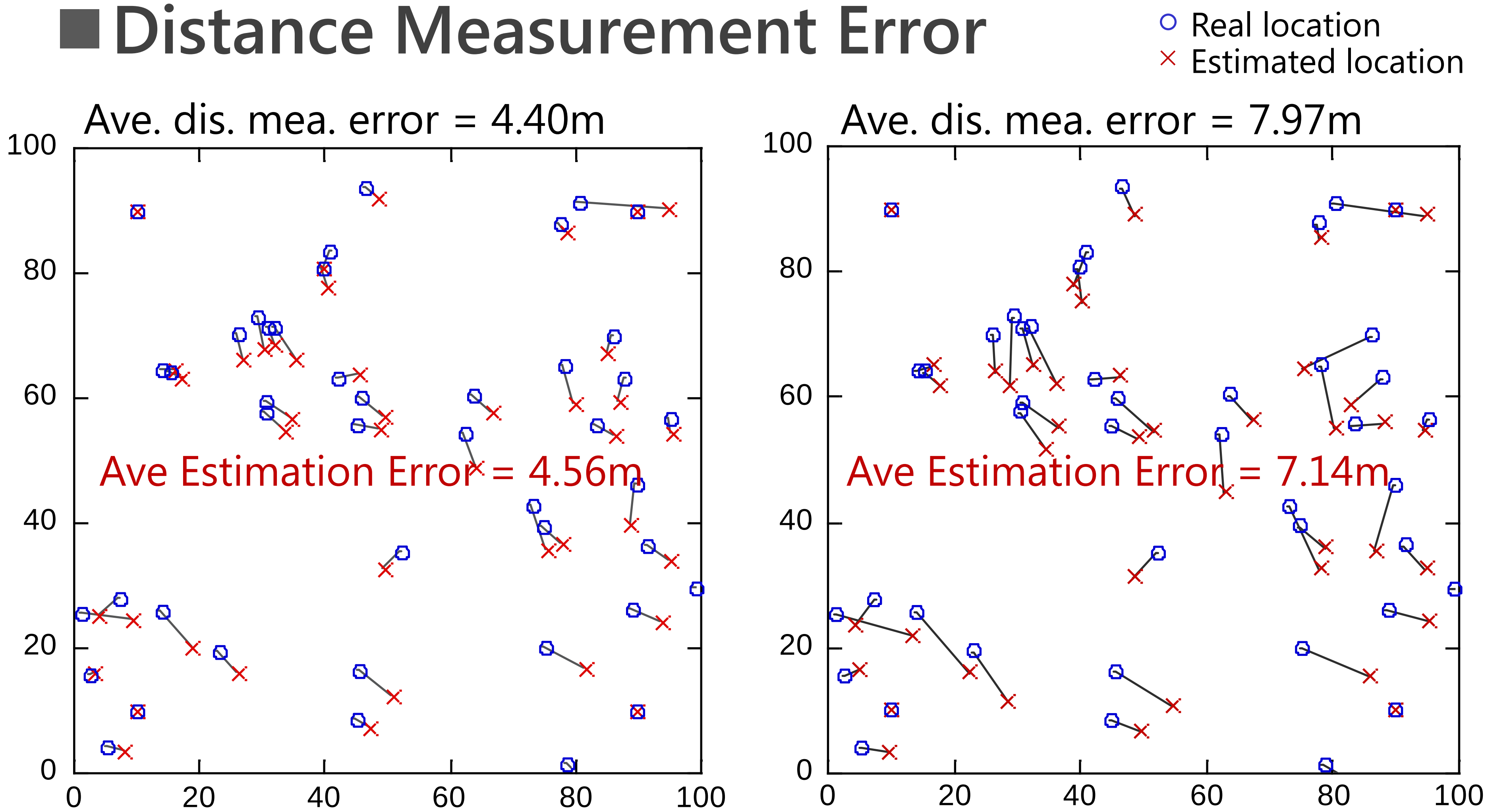
N_σ: Random variable (Noise) following Gaussian distribution

η: path loss exponent (assumed to be 4)

σ: standard deviation of RSS measurement (assumed to be 4 or 7)

Numerical Example

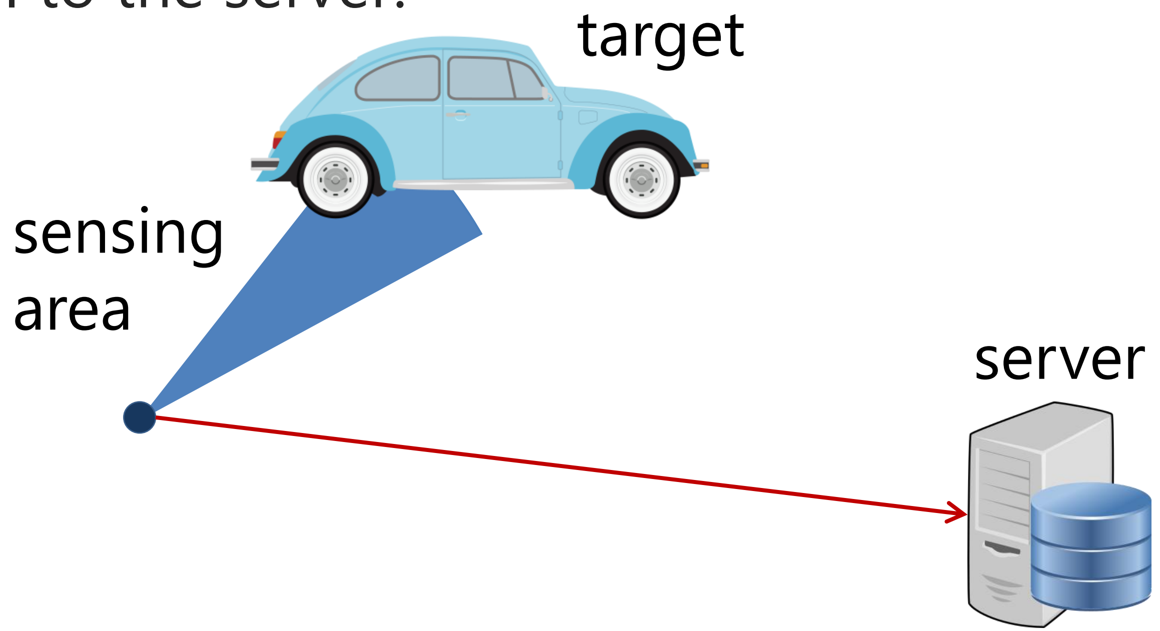
Distance Measurement Error



Localization is resilient to distance measurement errors.

Target Detection Sensor

- **Target Detection Sensor**
  - Detects a target when the target is in its sensing area.
  - Example: infrared sensor, which emits infrared (IR) rays and detects the amount of IR light that returns.
  - When detecting a target, it is assumed to send a signal to the server.



Localization via Responses to Targets

Localization via Responses to Targets

- Localize sensors based only on the responses to target

Time	Sensor ID
July 6, 10.20.15	2, 10, 110
July 6, 10.20.20	9, 38, 87
July 6, 10.20.25	19, 72, 101
July 6, 10.20.30	37, 73

- Location of a target is not known.

Idea

- Responses of sensors to a target tells us the spatial relationship between sensors.
- For example, two sensors should be located in close proximity when they detect a target at the same time.

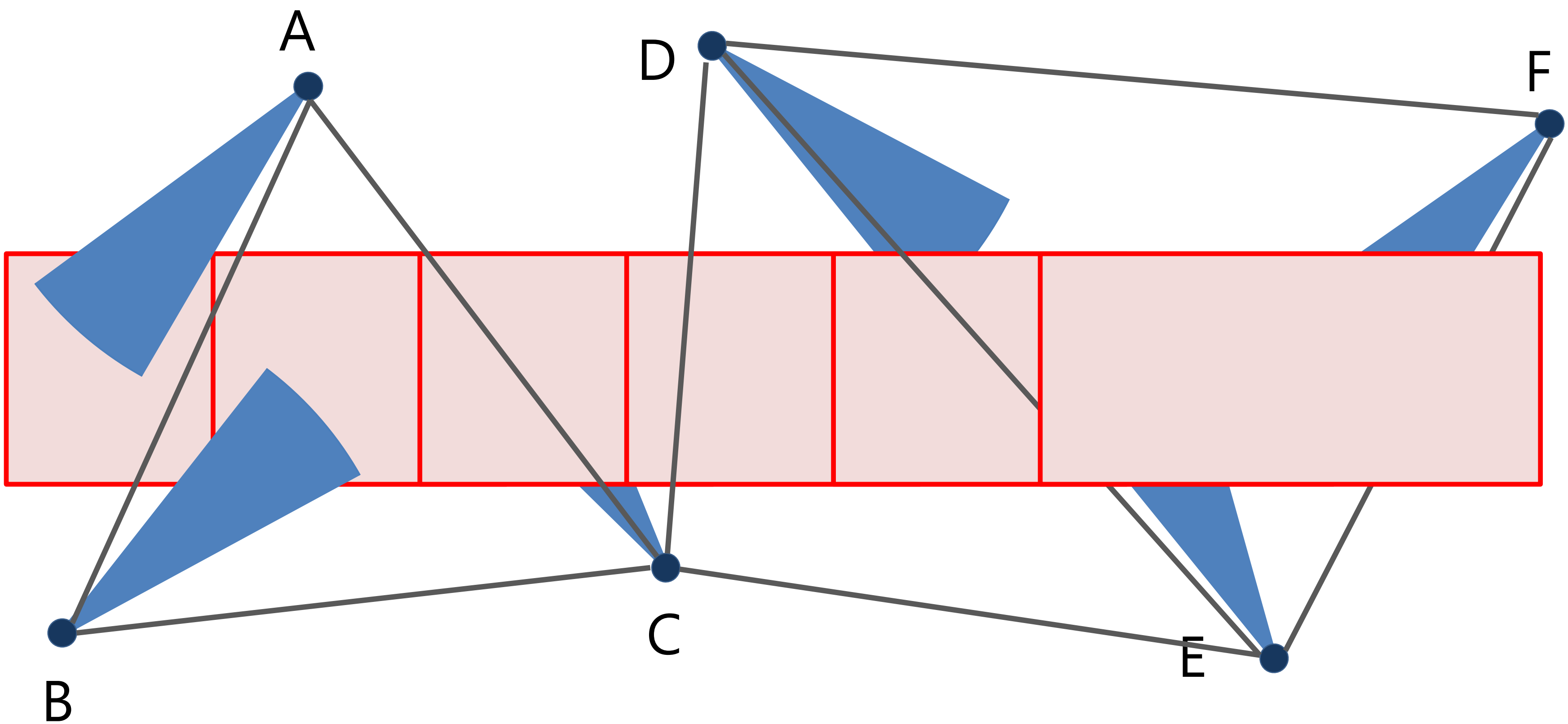
Localization via Responses to Targets

Localization Procedure

- Assume that two sensors are connected when they detect a target at the same time.
- From responses of sensors to targets, get connectivity information between sensors using the above assumption.
- Apply range-free (connectivity-based) localization technique.

Localization via Responses to Targets

- Example

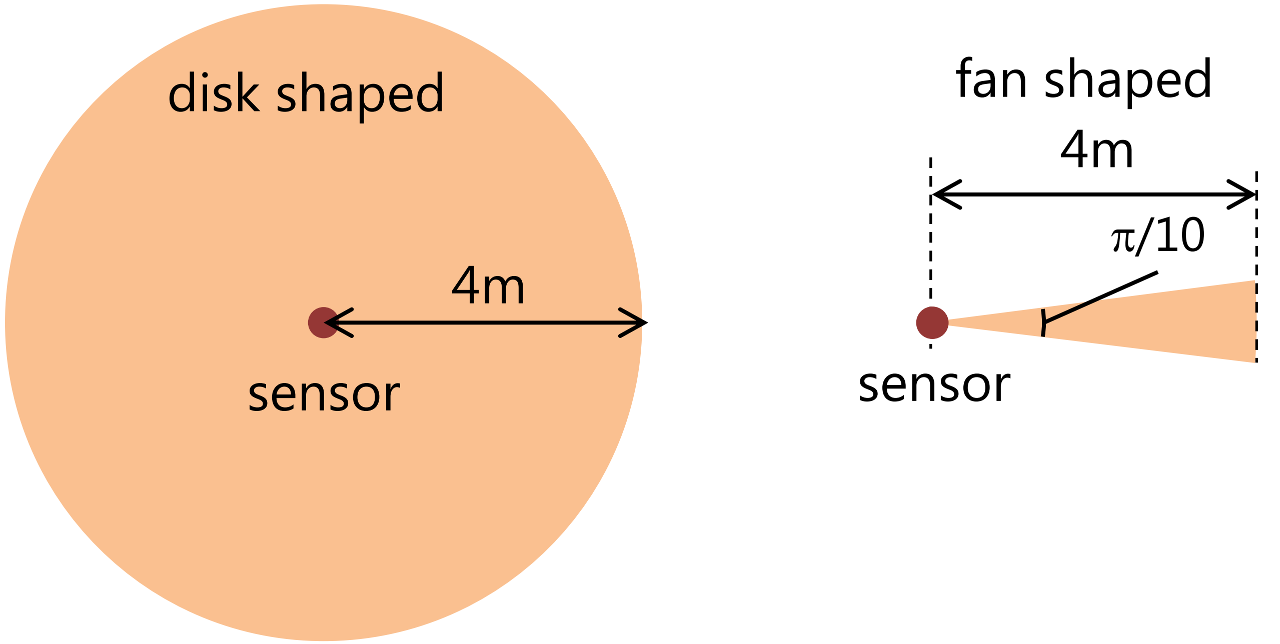


We get connectivity information!

Numerical Example

■ Simulation Condition 1

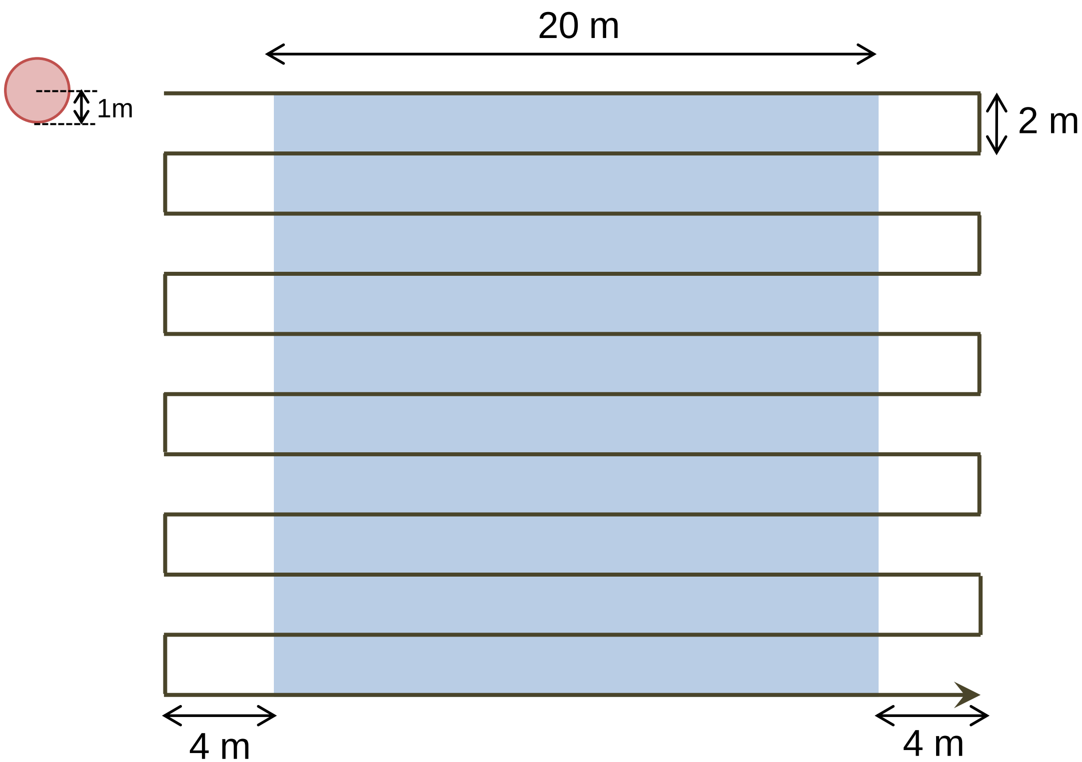
- Sensors are randomly deployed in a field (20m x 20m) with density of 0.5 [1/m^2].
- 4 anchors with known locations placed in corners.
- Each sensor has a disk- or fan-shaped sensing area.



Numerical Example

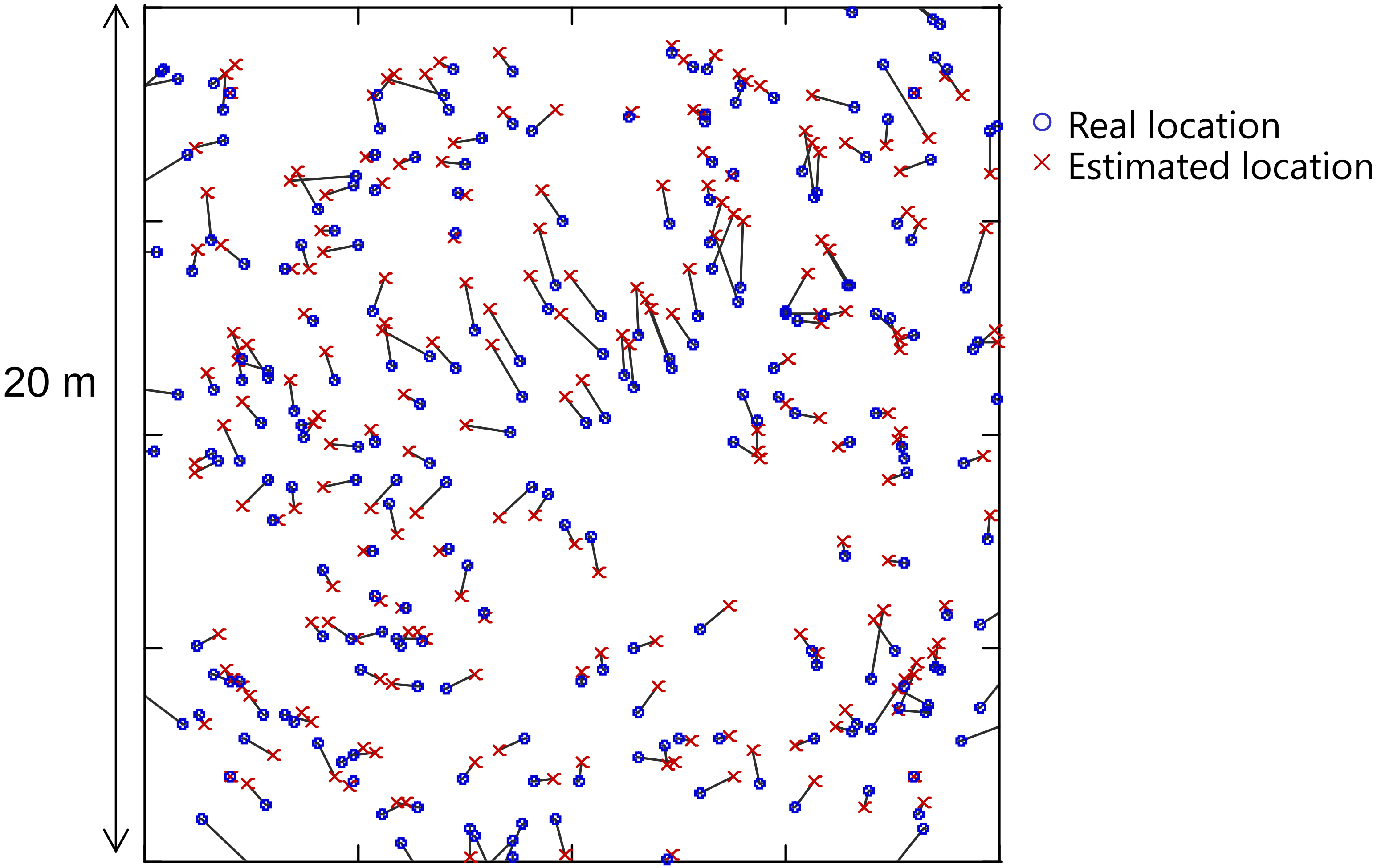
■ Simulation Condition 1 (cont.)

– Disk shaped object moves on the field along the trajectory shown below.



Numerical Example

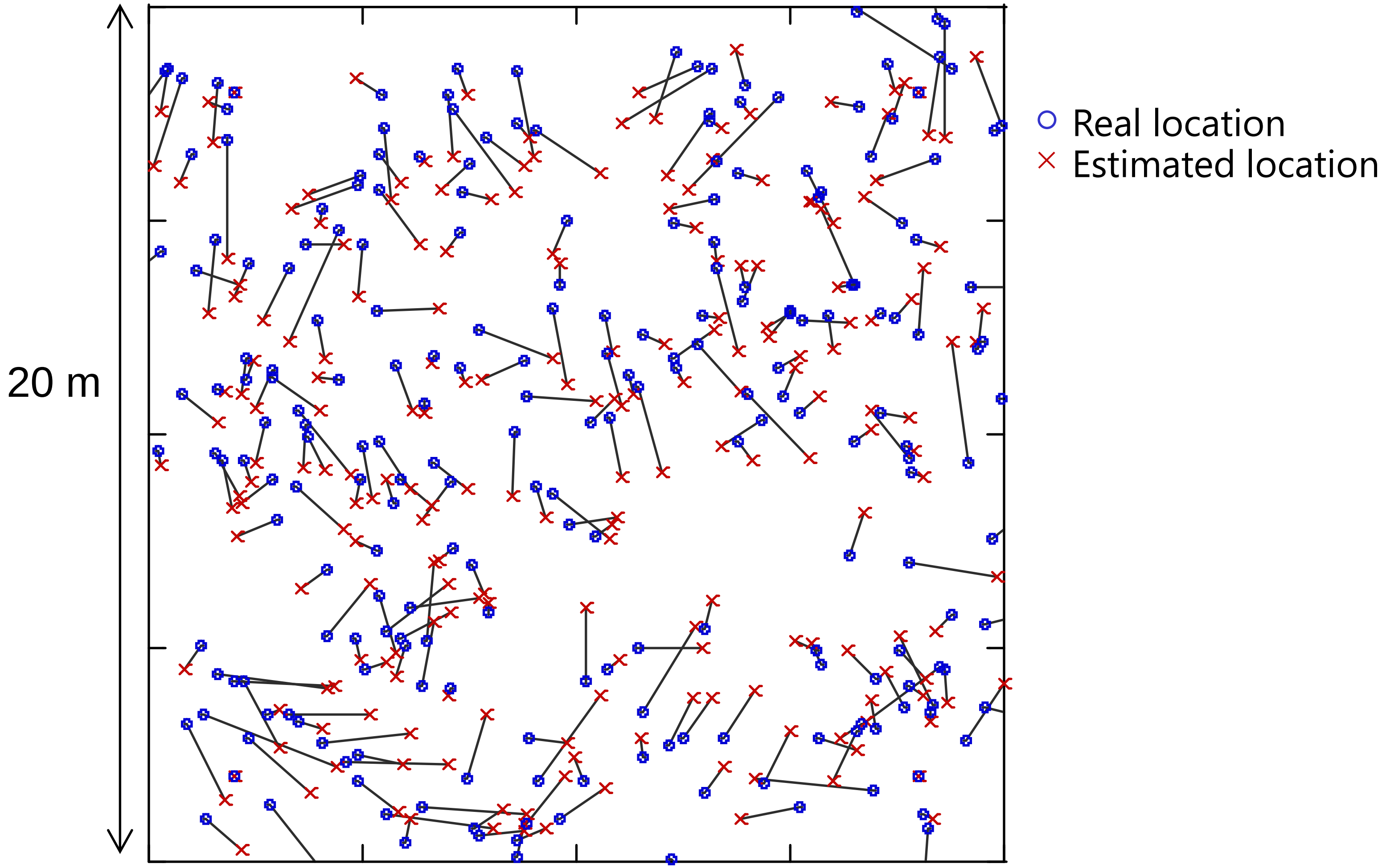
■ Results (disk-shaped sensing area)



Ave Estimation Error = 0.71m

Numerical Example

■ Results (fan-shaped sensing area)

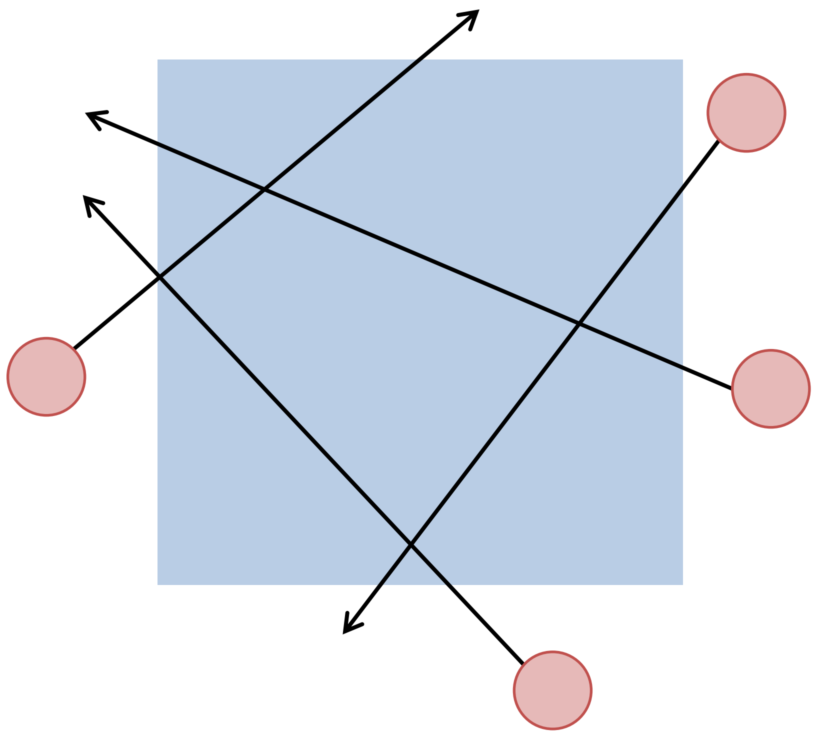


Ave Estimation Error = 1.15m

Numerical Example

■ Simulation Condition 2

- Disk-shaped objects with radius of 1 m traverse the field along (randomly chosen) straight lines one by one.
- See how a sensor location map has gradually been built up.

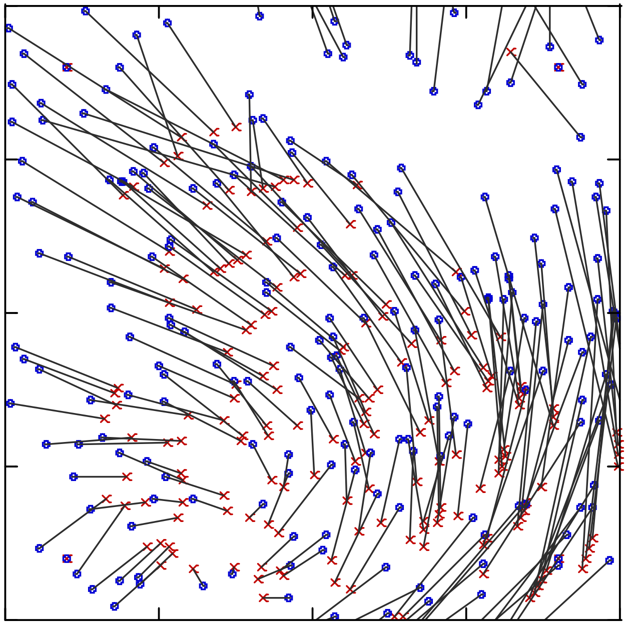


Numerical Example

■ Results (disk-shaped sensing area)

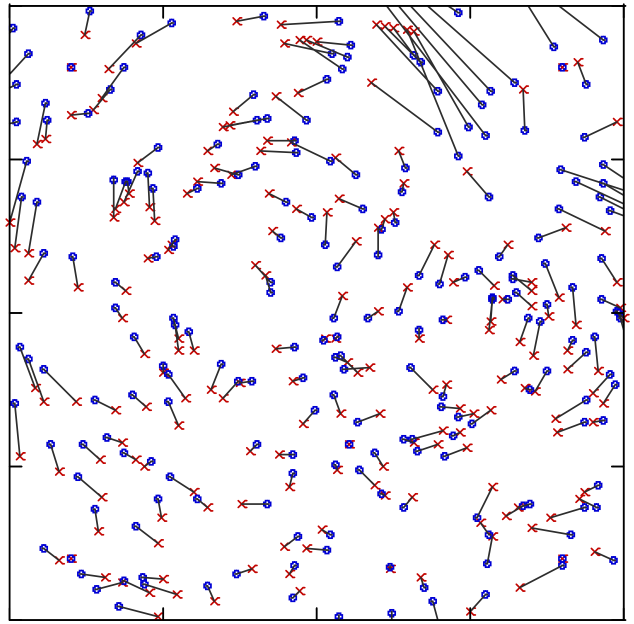
○ Real location
× Estimated location

10 objects have passed



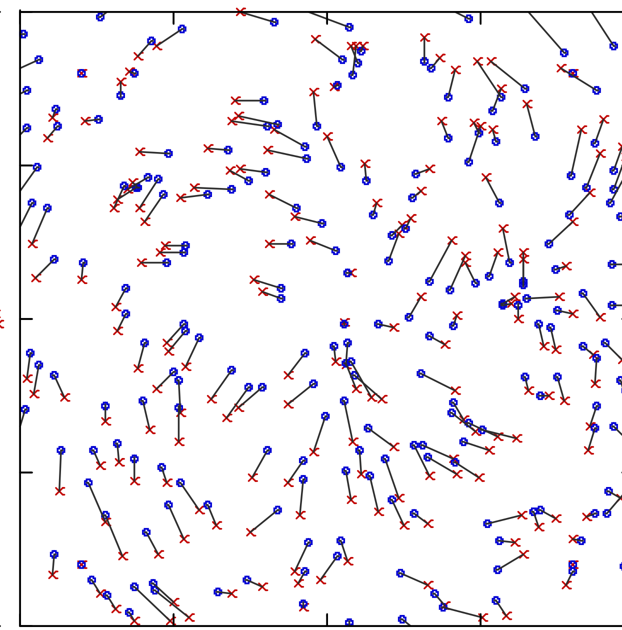
Ave Estimation Error = 2.45m

20 objects have passed



Ave Estimation Error = 1.13m

30 objects have passed



Ave Estimation Error = 0.88m

Sensor location map is gradually built up!

Summary & Future Work

■ Summary

– Distance-matrix (DM) based sensor localization is very promising.

– Applicable to either range-based or range-free method

– A wide variety of application ("localization based on the sensor responses to targets" is one application)

■ Future work

– Verification via experiments using real sensors

– Compressed-sensing (sparse-modeling) based localization

(Comparison of pros and cons of compressing-sensing- and DM-based localizations)

Reference

1. S. Shioda, "Localizing sensors from their responses to targets," IEICE Trans. Commun., vol. E98-B, no. 1, pp. 145-152, 2015.

2. S. Shioda, J. Komatsu, and K. Nishihara, "Connectivity-based sensor localization for anisotropic networks by Stress Relaxation," IEEE VTC Fall, 2015.

3. S. Shioda and K. Shimamura, "Relative localization of sensors based on their responses to moving objects," IEEE MASS, 2013 (IEEE MASS 2013 Best Poster Award).

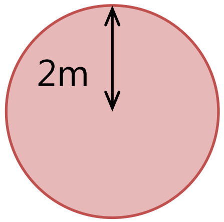
4. S. Shioda and K. Shimamura, "Cooperative localization revisited: error bound, scaling, and Convergence," ACM MSWiM'13, 2013.

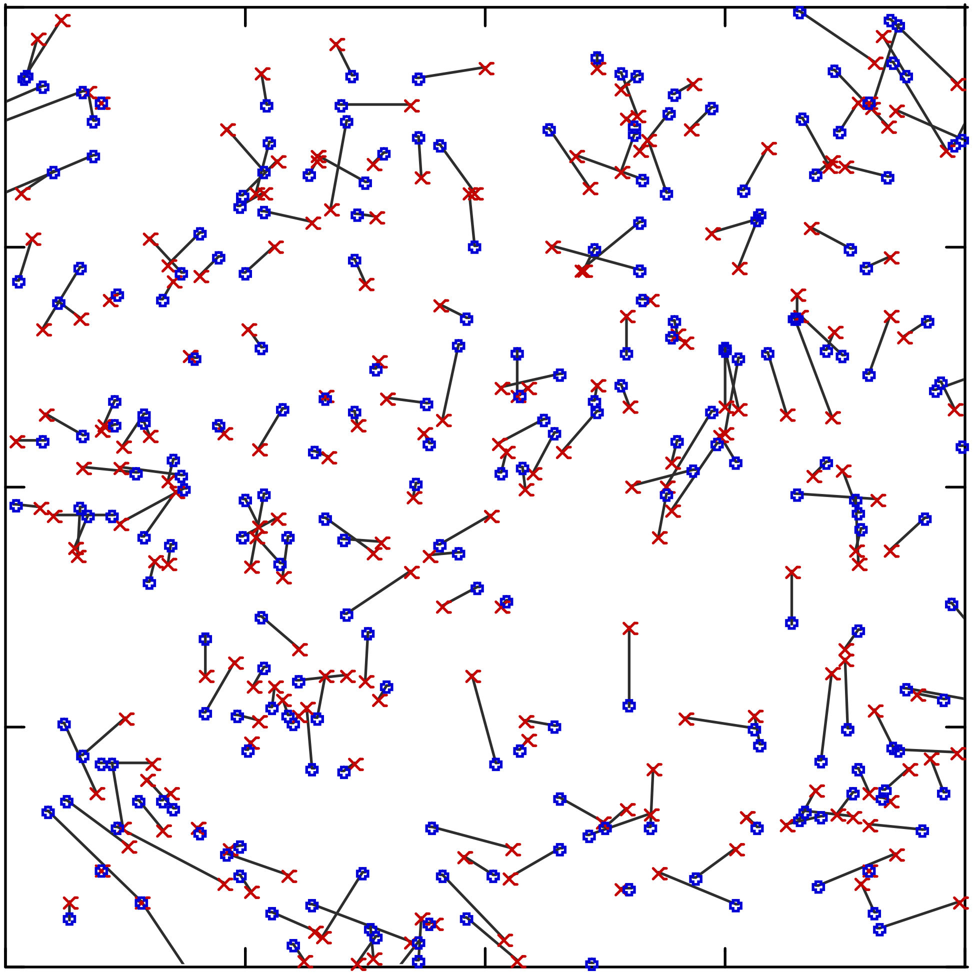
5. S. Shioda and K. Shimamura, "Anchor-free localization: estimation of relative locations of sensors," IEEE PIMRC, 2013.

Thank you very much!

Numerical Example

■ Results



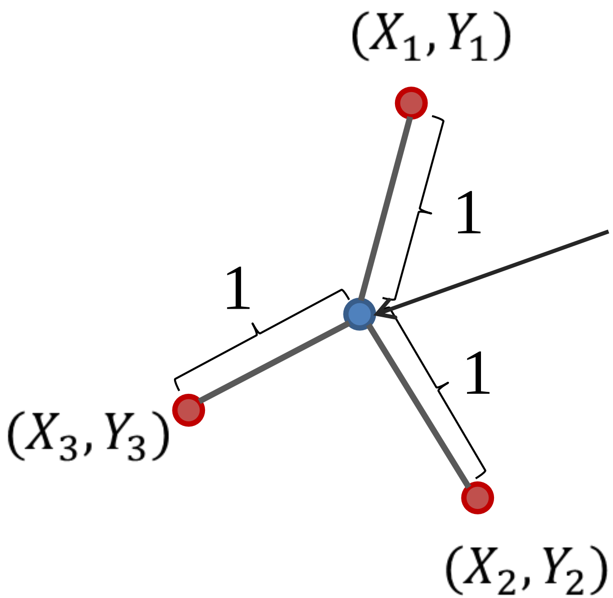


Ave Estimation Error = 0.94m

Range-Free Localization

Example: Centroid Algorithm

- Estimates the location as the centroid of anchors in the neighborhood.
- Note: it is equivalent to a range-based localization, where distances to anchors are equal to 1.



centroid of anchors

$$\left(X = \frac{X_1 + X_2 + X_3}{3}, Y = \frac{Y_1 + Y_2 + Y_3}{3}\right)$$

● anchors (with known location)

Range-Free Localization

- **Example 2: DV-Hop Algorithm**
  - Extension of the centroid Algorithm
  - It is equivalent to a range-based localization, where distance to an anchor is given as the number of hops