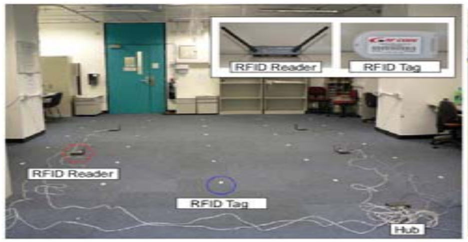
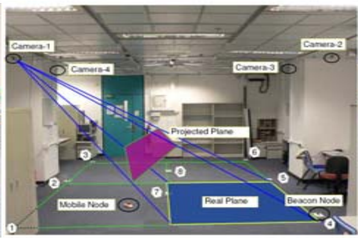
Learning-based Localization in Wireless and Sensor Networks

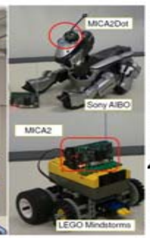
Jeffrey Junfeng Pan

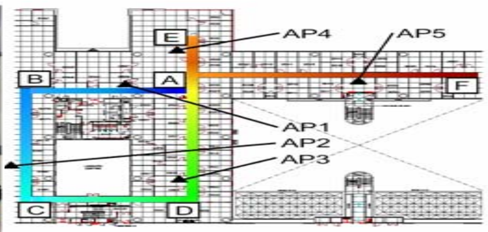
Advisor: Qiang Yang

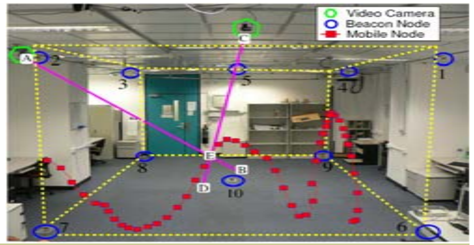
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology





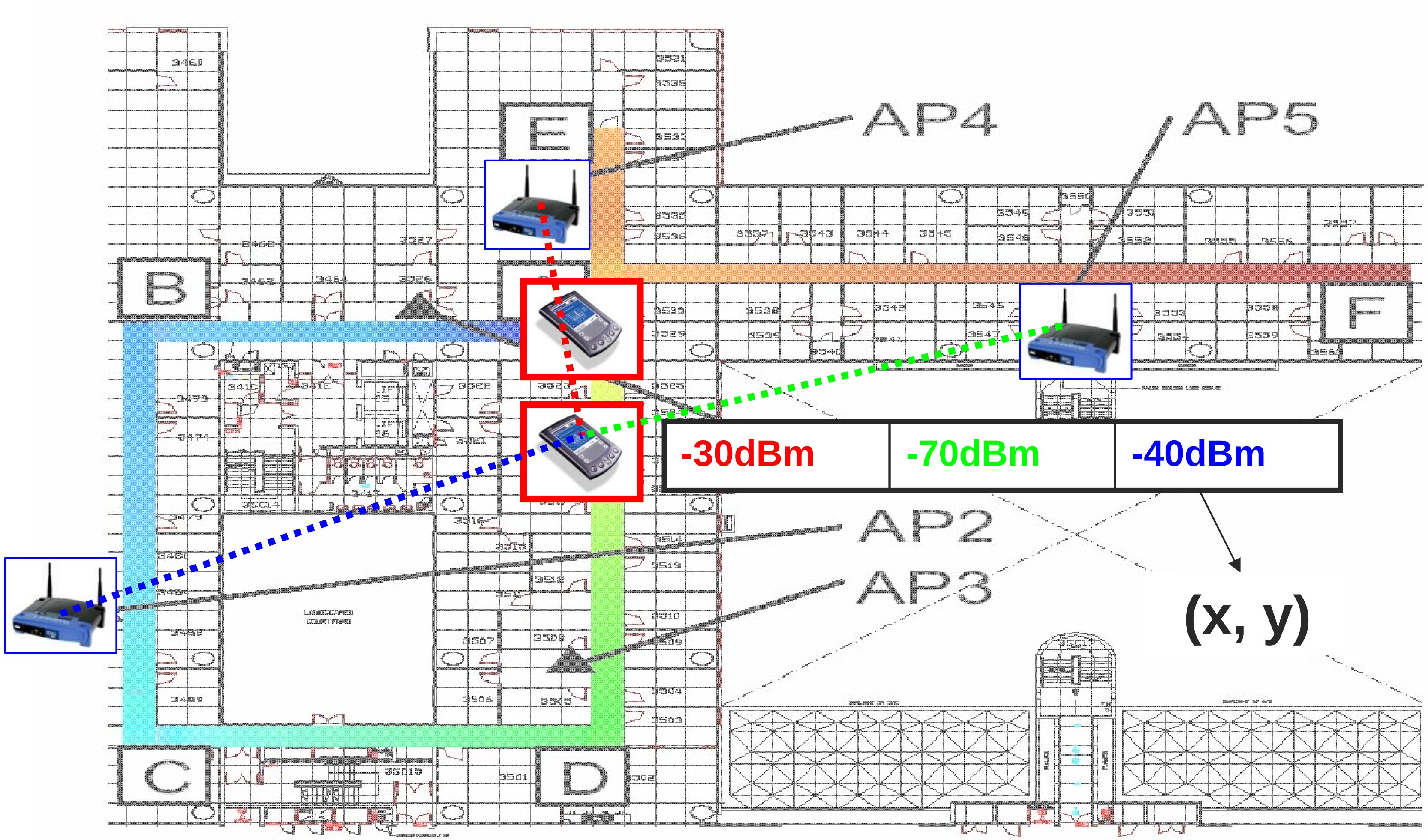






Signal-Strength-Based Localization
Where is the Mobile Device?

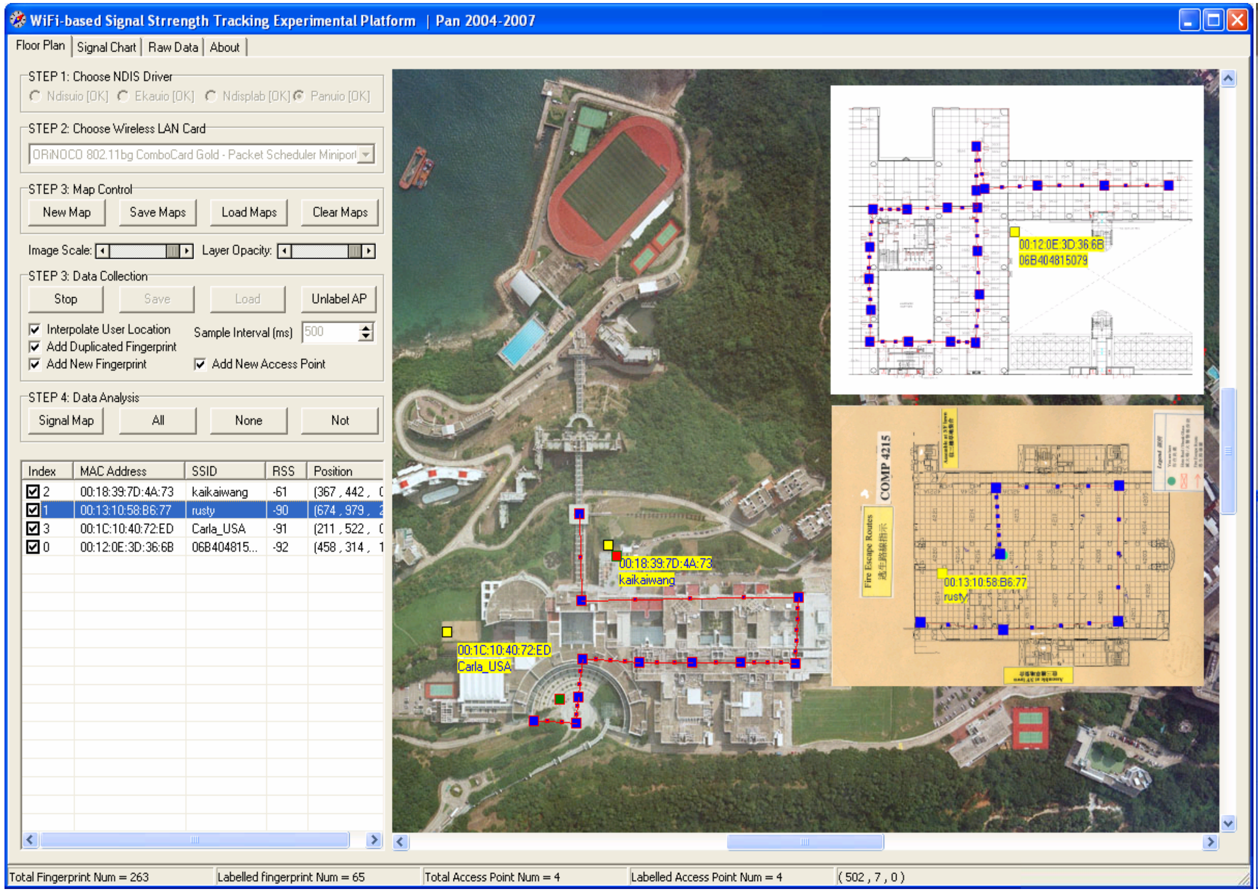
HKUST

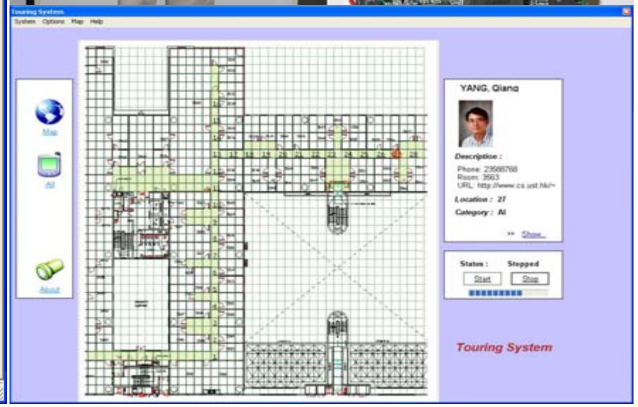


Locations Support Many Applications

HKUST

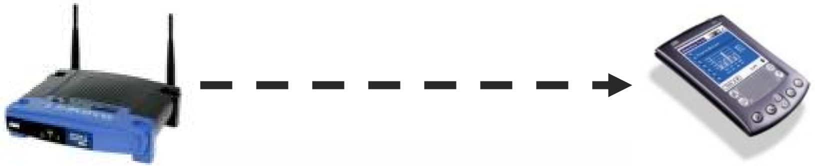
- Guidance, Content Delivery & User Behavior Analysis

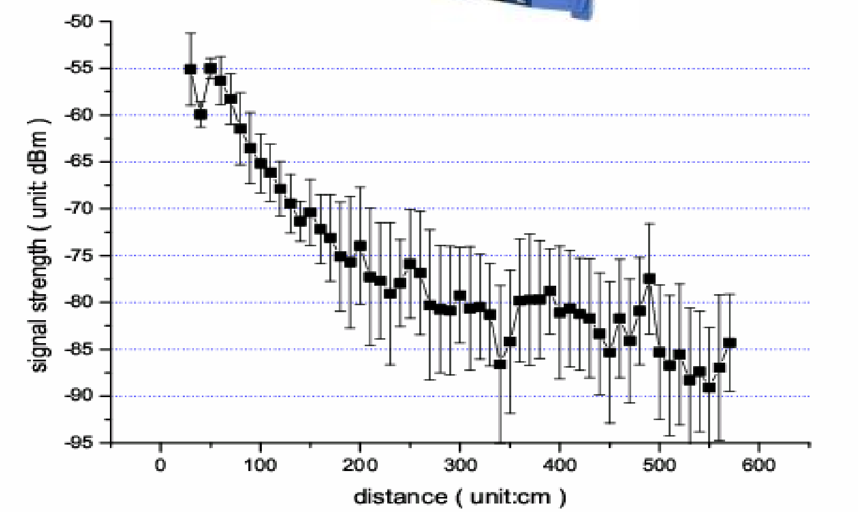




“Cheap and Ubiquitous Received-Signal-Strength” ?

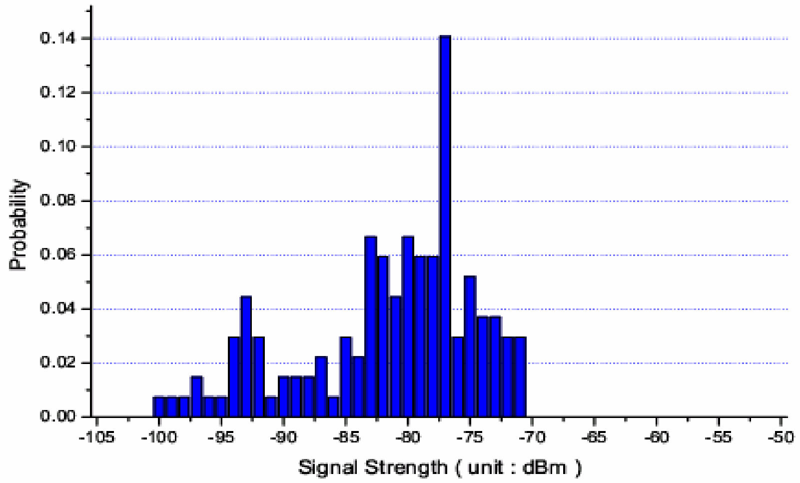
HKUST





Signal attenuation along with distance

nonlinear



Signal Distribution at a fixed location

noisy

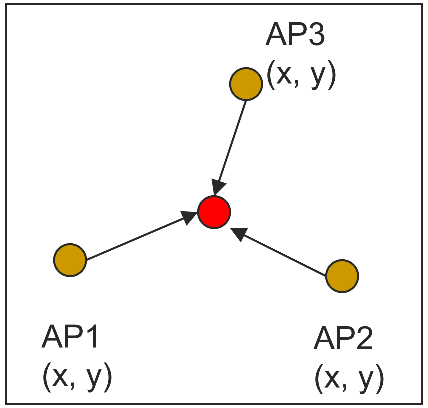
Related Works

HKUST

- Radio Propagation Models – (**Rely on AP Locations**)
  - Log-Normal model -- Bayesian indoor positioning systems. Maligan et al. INFOCOM 2005.
  - Multilateration – Dynamic fine-grained localization in ad-hoc networks of sensors. MOBICOM 2001

- Machine Learning Models – (**Not Need AP Locations**)
  - KNN -- LANDMARC: Indoor Location Sensing Using Active RFID. Ni et al. PerCom 2003. / RADAR etc.
  - ML -- Large-scale localization from wireless signal strength. Letchner et al. AAAI 2005.
  - SVM -- A kernel-based learning approach to ad hoc sensor network localization. Nguyen et al. ACM Transaction on Sensor Networks. 2005.

Propagation-based Models



- Path Loss [Goldsmith et al.]
  - power radiation

$$P(d)[dBm] = P(d_0)[dBm] - 10n\log\frac{d}{d_0}$$

- Shadowing [Maligan et al.]
  - absorption, reflection, scattering, and diffraction

$$P(d)[dBm] = P(d_0)[dBm] - 10n\log\frac{d}{d_0} - \Phi(d)$$

- Indoor Attenuation Factors [Bahl et al.]
  - floors and walls

$$P(d)[dBm] = P(d_0)[dBm] - 10n\log\frac{d}{d_0} - \Phi(d) - \sum_{i=1}^{N_f} FAF_i - \sum_{i=1}^{N_p} PAF_i$$

- Multipath [Goldsmith et al.]
  - ray-tracing, need more detail about environment

Question One

Can we build an accurate mapping from signals to locations?

* It is not easy to parameterize an indoor environment, (wall material, building structure, etc.)

Learning-based Models

- Two phases: 1) offline Training and 2) online Localization
- **Offline phase** – collect data to build a mapping function F from signal space **S**(AP1,AP2,AP3) to location space **L**(x,y)

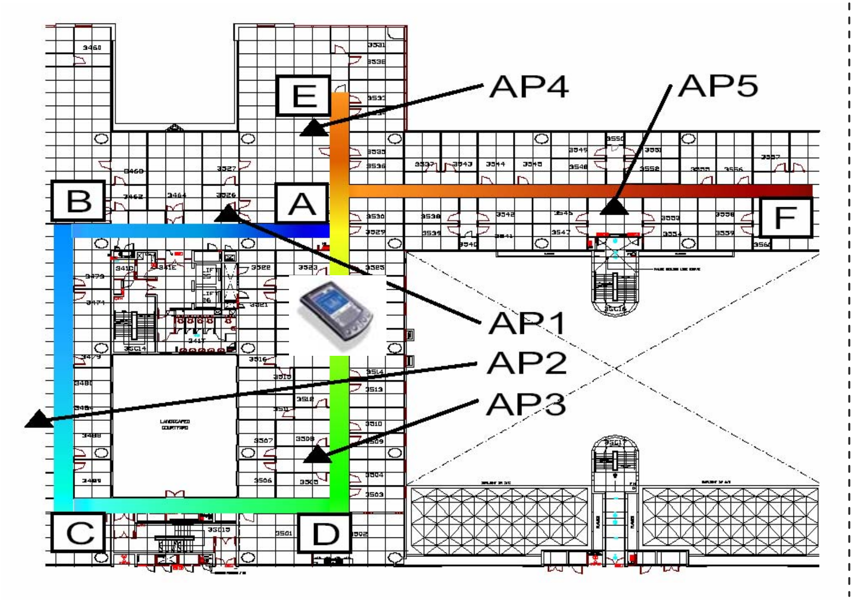
Time	(AP1, AP2, AP3)	(x, y)
T_1	(-60,-50,-40) dB	(1,0)
T_2	(-62,-48,-35) dB	(2,0)
….	(… , … , …) dB	…..
T_N	(-50,-35,-42) dB	(9,5)

Training → Mapping function F

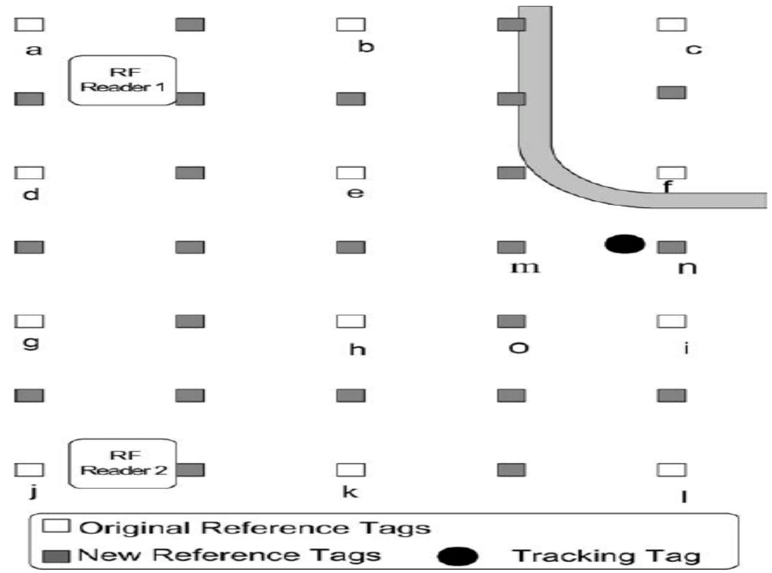
- **Online phase** – given a new signal s , estimate the most likely location l from function F
  - s = (-60,-49,-36) dB , compute l=F(s) as the estimated location

Learning-based Models (cont')

- Manually Setup
  - HORUS/RADAR/…
  - Walk to several points
    Collect data manually
  - Cost time



- Semi-automatically Setup
  - LANDMARC
  - Mark tag position manually
    Collect data automatically
  - Cost money



Question Two

Can we reduce calibration effort?

* We need to collect a lot of data at many locations

Question Three & Four

HKUST

Can a learning-based model benefit if access points are calibrated?

*Propagation models use AP locations while learning models don't.

Can a learning-based model work purely online for adaptation?

*Learning models usually function in two phases: offline/online

Our Contribution

HKUST

- Localization Models (a general framework)
  - A Flexible Model for Localization and Mapping
  - Increase accuracy with known-location clients or APs
  - Reduce calibration with unknown-location clients or APs
  - Can work offline/online or purely online for adaptation

- Localization Experiments (thorough study)
  - Devices: WiFi, Sensor Networks, RFID Networks
  - Test-bed: Hallways, Indoor open space, 2D & 3D
  - Mobility: Static, Moving persons, Moving robots.

Question One

Can we increase the accuracy when some labelled* (calibrated) data are available?

*A Labelled (calibrated) example is an input/output pair
Example: (-60dBm,-50dBm,-70dBm) => (x,y)

Observation of Signal Strength

HKUST

- A user with a mobile device walks through A B,C,D,E,F

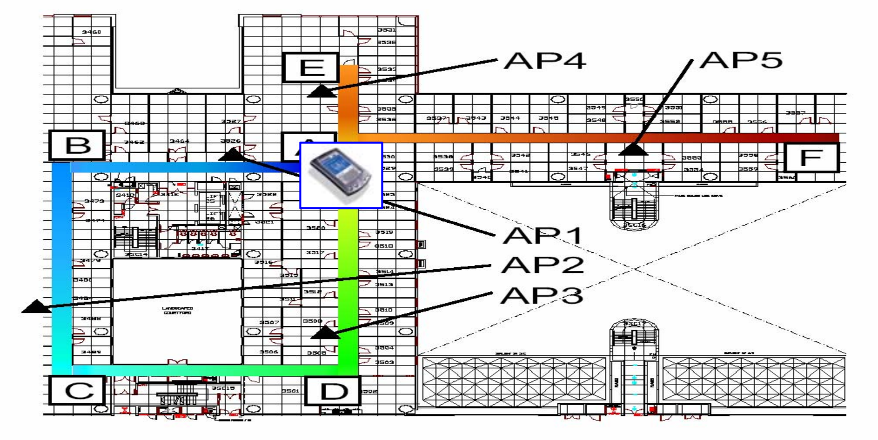


Table 1: Signal Strength (unit:dBm)

		AP_1	AP_2	AP_3	AP_4	AP_5
(x1,y1)	t_A	-40		-60	-40	-70
(x2,y2)	t_B	-50	-60		-80	
(x3,y3)	t_C		-40	-70		
(x4,y4)	t_D	-80		**-40**	-70	
(x5,y5)	$t_{A'}$	-40		-70	-40	-60
(x6,y6)	t_E	-40		-70	-40	-80
(x7,y7)	t_F	-80			-80	-50

(All values are rounded for illustration)

- Characteristics (statistically)
  - Two rows are similar ⇔ Two mobile devices are close (t_A & $t_{A'}$)
- However, when observing individual noisy data points
  - Similar signals may not be nearby locations
  - Dissimilar signals may not be far away

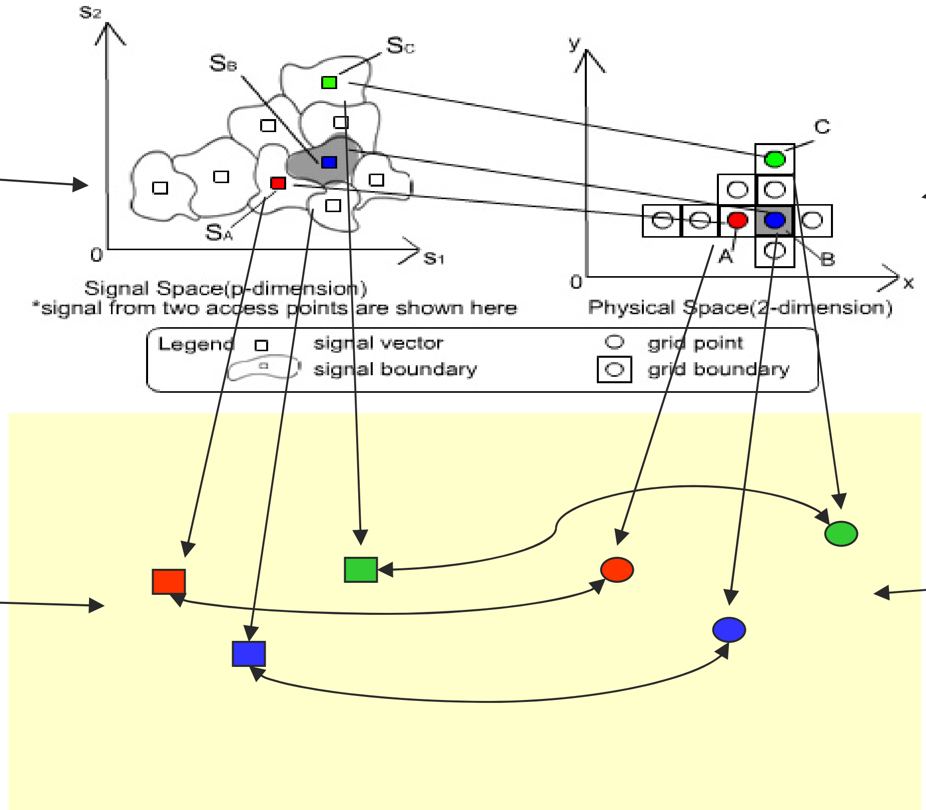
Motivation of Our Approach

HKUST

Idea: Maximize the similarity correlation between signal and location spaces

Original Signal Space

Original Location Space



Feature Signal Space

Feature Location Space

(Kernel) CCA

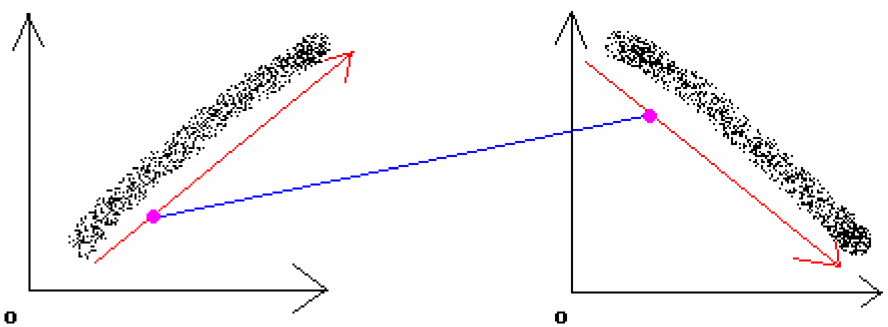
HKUST

- **C**anonical **C**orrelation **A**nalysis (CCA)
  - [H. Hotelling, 1936]
  - Two data set X and Y
  - Two linear Canonical Vectors **Wx Wy**
  - Maximize the correlation of projections

$$S_{x,\mathbf{w}_x} = (\langle \mathbf{w}_x, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{w}_x, \mathbf{x}_n \rangle)$$

$$S_{y,\mathbf{w}_y} = (\langle \mathbf{w}_y, \mathbf{y}_1 \rangle, \ldots, \langle \mathbf{w}_y, \mathbf{y}_n \rangle)$$

$$\begin{aligned} \rho &= \max_{\mathbf{w}_x, \mathbf{w}_y} corr(S_x \mathbf{w}_x, S_y \mathbf{w}_y) \\ &= \max_{\mathbf{w}_x, \mathbf{w}_y} \frac{\langle S_x \mathbf{w}_x, S_y \mathbf{w}_y \rangle}{\|S_x \mathbf{w}_x\| \|S_y \mathbf{w}_y\|} \end{aligned}$$



- **Kernel CCA**
  - [D.R Hardoon, S. Szedmak, and J. Shawe-Taylor, 2004]
  - Two non-linear Canonical Vectors

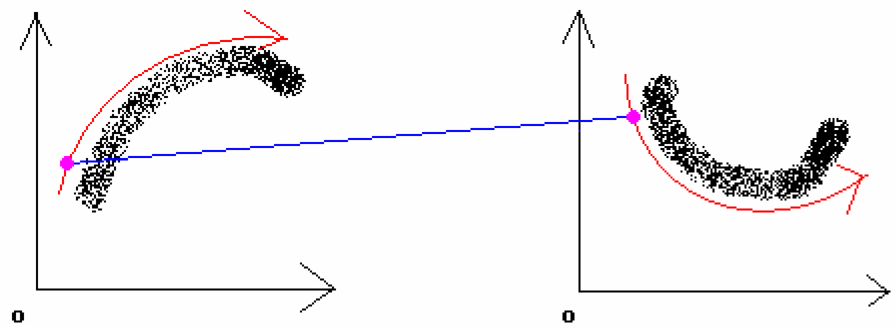
$$\mathbf{w}_x = X\alpha \qquad \mathbf{w}_y = Y\beta$$

  - K is the kernel

$$\phi : \mathbf{x} \mapsto \phi(\mathbf{x})$$

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle$$

$$\rho = \max_{\alpha, \beta} \frac{\alpha' K_x K_y \beta}{\sqrt{\alpha' K_x^2 \alpha \cdot \beta' K_y^2 \beta}}.$$



LE-KCCA

HKUST

- **Offline phase**
  - Signal strengths are collected at various grid locations.
  - KCCA is used to learn the mapping between signal and location spaces.
    - λ_i's and α_i's are obtained from the generalized eigen-problem
    - κis a regularization term

$$(\mathbf{K_x}+\kappa\mathbf{I})^{-1}\mathbf{K_y}(\mathbf{K_y}+\kappa\mathbf{I})^{-1}\mathbf{K_x}\alpha=\lambda^2\alpha,$$

  - For each training pair ($\mathbf{s}_i$, l_i), its projections

$$P(\mathbf{s}_i)=[P_1(\mathbf{s}_i),P_2(\mathbf{s}_i),\ldots,P_T(\mathbf{s}_i)]'$$

on the *T* canonical vectors are obtained from

$$P_{\mathbf{x}}(\tilde{\mathbf{x}})=\phi_x(\tilde{\mathbf{x}})'\mathbf{w}_{\phi_x(\mathbf{x})}=\mathbf{k}'_{\tilde{\mathbf{x}}}\alpha,$$

LE-KCCA (Cont')

- **Online phase**
  - Assume the location of a new signal strength vector is **s**
  - Again, use

    $$P_{\mathbf{x}}(\tilde{\mathbf{x}}) = \phi_x(\tilde{\mathbf{x}})' \mathbf{w}_{\phi_x(\mathbf{x})} = \mathbf{k}'_{\tilde{\mathbf{x}}} \alpha,$$

    to project **s** onto the canonical vectors and obtain

    $$P(\tilde{\mathbf{s}}) = [P_1(\tilde{\mathbf{s}}), P_2(\tilde{\mathbf{s}}), \ldots, P_T(\tilde{\mathbf{s}})]'.$$

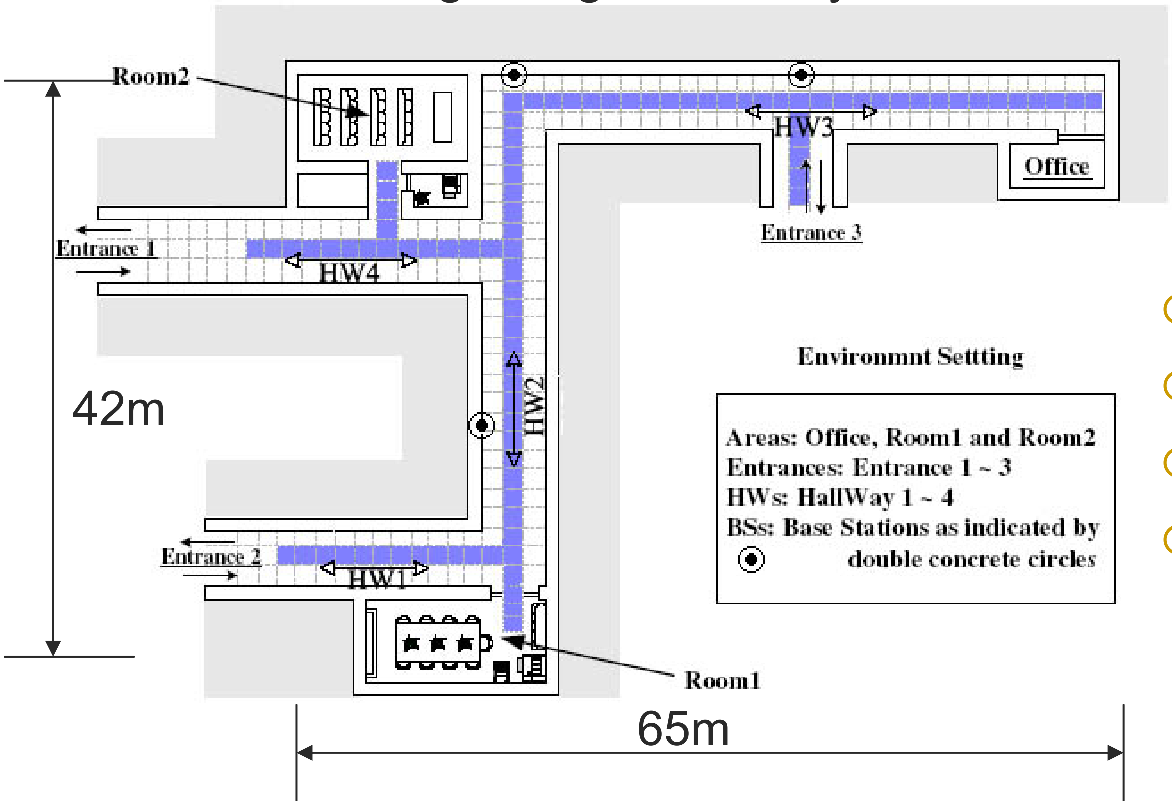
  - Find the K *Nearest N*eighbors of *P*(**s**) in the projections $P(\mathbf{s}_i)$ of training set with the weighted Euclidean distance :

    $$d_i = \sum_{j=1}^{T} \lambda_j (P_j(\tilde{\mathbf{s}}) - P_j(\mathbf{s}_i))^2$$

  - Interpolate these neighbors' locations to predict the location of **s**
- Essentially, we are performing Weighted KNN in the feature space with which weights are obtained from the feedback of location information.

Experimental Setup

- Test-bed
  - Department of Computer Science and Engineering
    Hong Kong University of Science and Technology

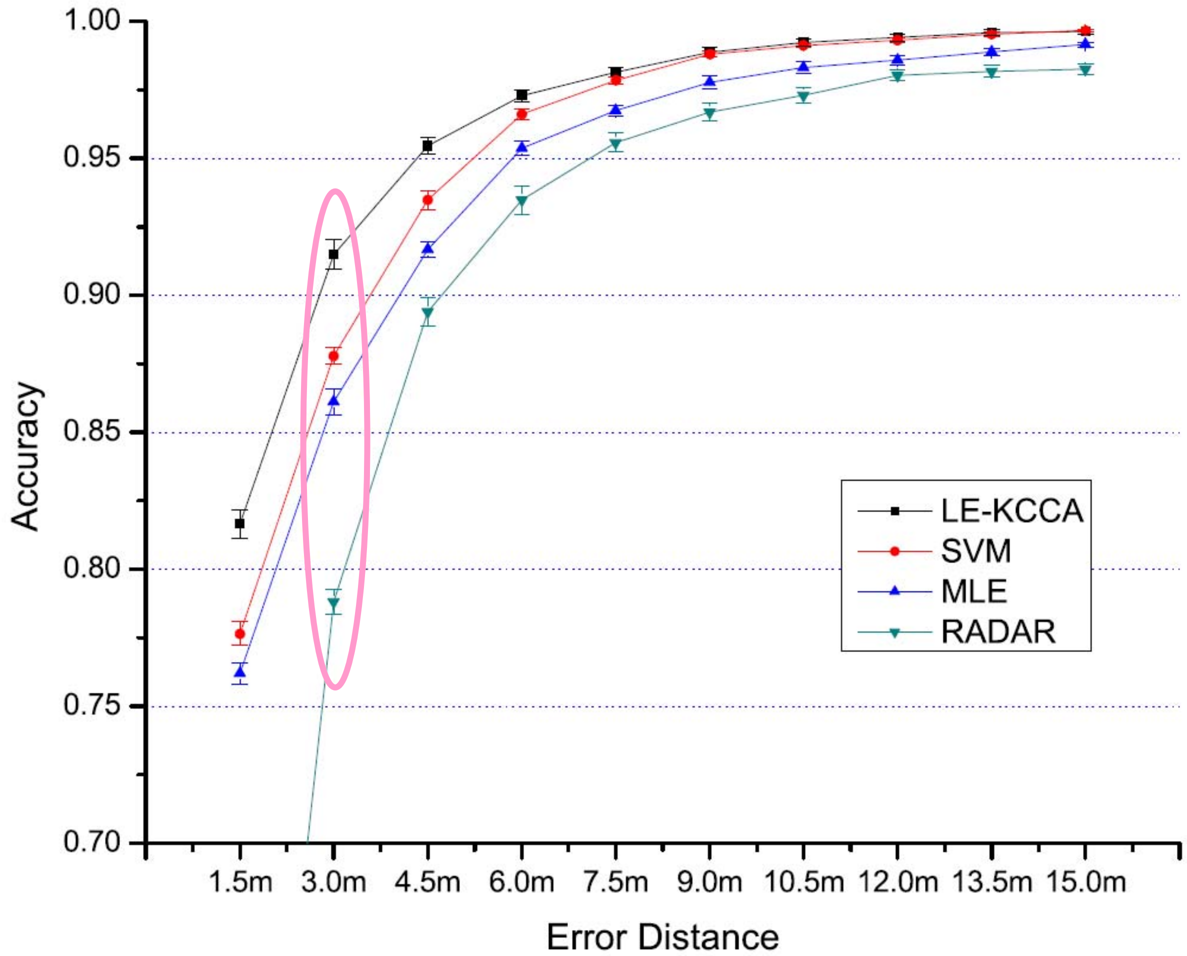


- 99 locations (1.5 × 1.5 meter)
- 100 samples per location
- 65% for training, 35% testing
- Repeat each experiment 10 times

Experimental Result

HKUST

- **How we use data set**
  - 65% training
  - 35% testing
  - 10 repetition
- **Error distance is 3.0m**
  - **LE-KCCA 91.6%**
  - **SVM 87.8%**
  - **MLE 86.1%**
  - **RADAR 78.8%**



Question Two

Can we reduce calibration effort by using additional unlabelled (uncalibrated) data?

- Labelled data are expensive to get
  - (-60dBm,-50dBm,-70dBm) => (x,y)
- Unlabelled data are easy to obtain
  - (-60dBm,-50dBm,-70dBm)

Observation of Signal Strength

HKUST

- A user with a mobile device walks through A B,C,D,E,F

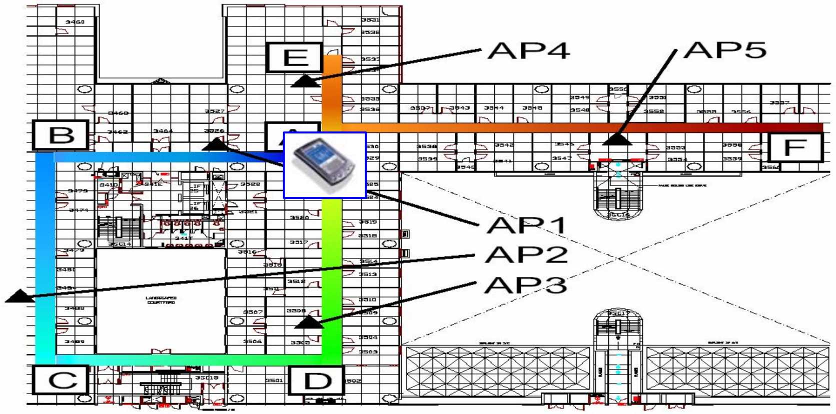


Table 1: Signal Strength (unit:dBm)

	AP_1	AP_2	AP_3	AP_4	AP_5
t_A	-40		-60	-40	-70
t_B	-50	-60		-80	
t_C		-40	-70		
t_D	-80		**-40**	-70	
$t_{A'}$	-40		-70	-40	-60
t_E	-40		-70	-40	-80
t_F	-80			-80	-50

(All values are rounded for illustration)

- Characteristics (statistically)
  - Two **rows** are similar ⇔ Two mobile devices are close (t_A & $t_{A'}$)
  - Neighbored **rows** are similar ⇔ User Trajectory is smooth (t_i & t_{i+1})
- Basic Idea
  - Bridge labelled and unlabelled data

Manifold Regularization

HKUST

- Basic Assumption
  - If two points are close in the intrinsic geometry (manifold) of the marginal distribution, their conditional distributions are similar
    - Classification – Similar Labels
    - Regression – Similar Values
- Infer the unlabelled data by
  - Taking a look at the neighbor points

Manifold Regularization (Cont')

- The Objective is to Optimize,

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l}\sum_{i=1}^{l} V(r_i, z_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2}\hat{f}^T L \hat{f},$$

Fitting Error (labelled data)	Function Complexity in Ambient Space	Function Complexity along Manifold (un/labelled)
Squared Fitting Error	**Squared Function Norm**	**L is the graph Laplacian**

$$V(r_i, z_i, f) = (z_i - f(r_i))^2$$

$$\|f\|^2 = \alpha' K \alpha$$

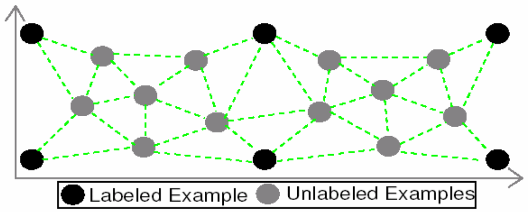


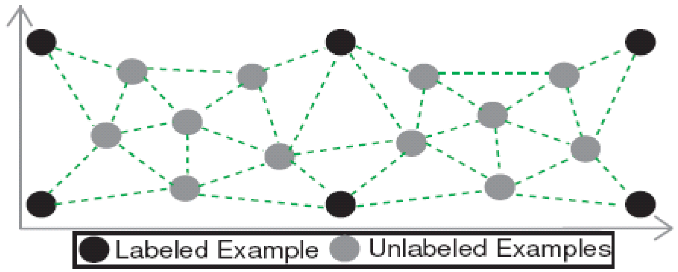
Figure 1: The use of labeled and unlabel examples

Optimal solution

$$\alpha^* = (JK + \gamma_A l I + \frac{\gamma_I l}{(u+l)^2} LK)^{-1} Z,$$

The *LeMan* Algorithm

HKUST



- Offline Training Phase
  - Collect l labeled and u unlabeled signal examples
  - Construct graph Laplacian L Kernel Matrix K
  - Solving for α

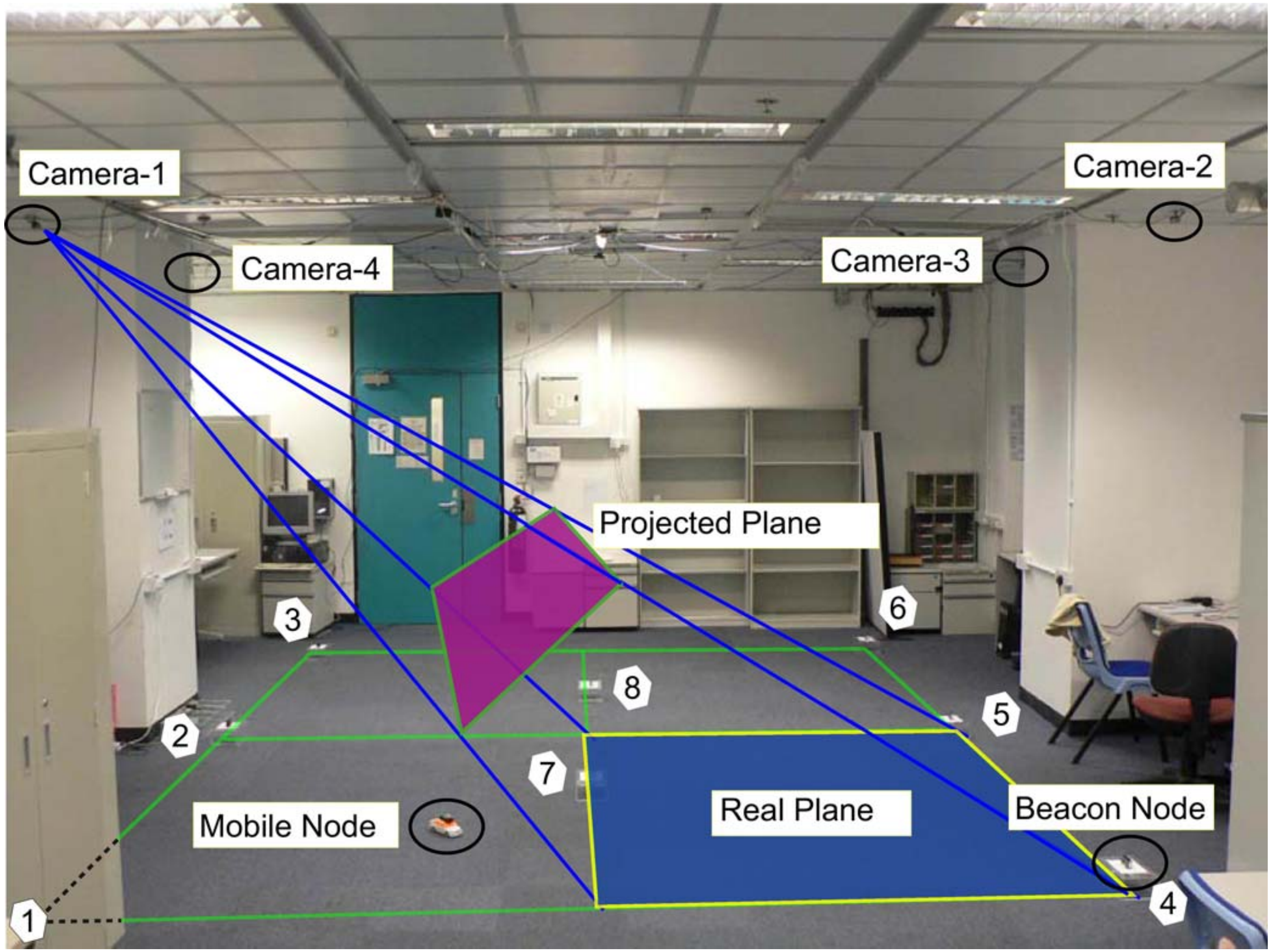
$$\alpha^* = (JK + \gamma_A lI + \frac{\gamma_I l}{(u+l)^2} LK)^{-1} Z,$$

- Online Localization Phase

$$f^*(r) = \sum_{i=1}^{l+u} \alpha_i K(r_i, r).$$

Online phase time complexity O((l+u)*N) where N is the number of sensors

Experimental Setup

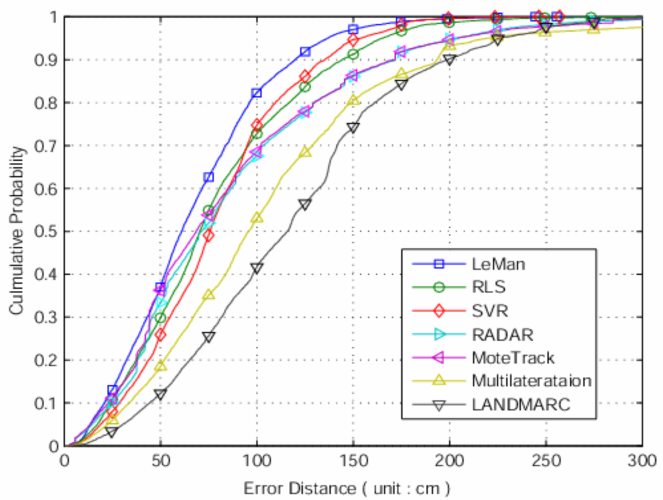


Location Estimation Accuracy

HKUST

Table 1: Performance of Different Methods

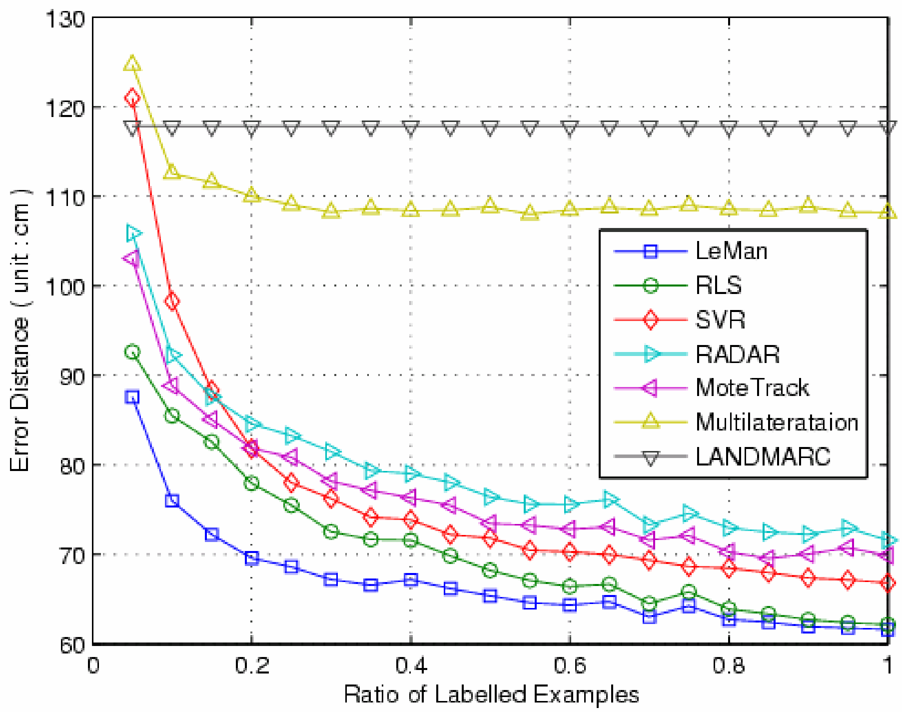
Method	Mean (cm)	Std. (cm)	Max (cm)	Accuracy at 100cm	Time (ms)
LeMan	* 67	* 39	290	* 82%	0.242
RLS	78	46	358	73%	0.047
SVR	79	40	* 257	75%	0.045
RADAR	86	59	391	68%	0.106
MoteTrack	85	61	418	69%	0.106
Multilateration	108	77	1592	53%	0.125
LANDMARC	118	59	372	42%	0.085



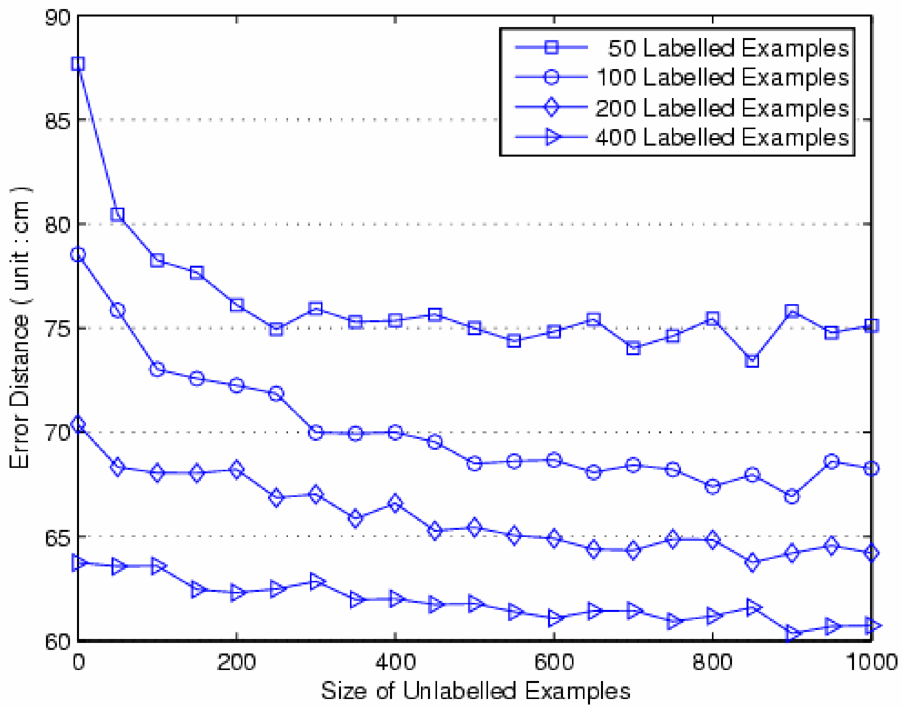
(a) Comparison of accuracy

- LeMan has the smallest mean error distance 67cm
- LeMan is robust. The std. of error distance is 39cm
- LeMan needs more computation time. 0.242ms per location on 3.2GHz CPU on Matlab.

Vary Labelled and Unlabelled Data



(e) Vary the ratio of labeled examples



(d) Vary the number of unlabeled examples

Question Three

Can we employ further information source, e.g., access point locations?

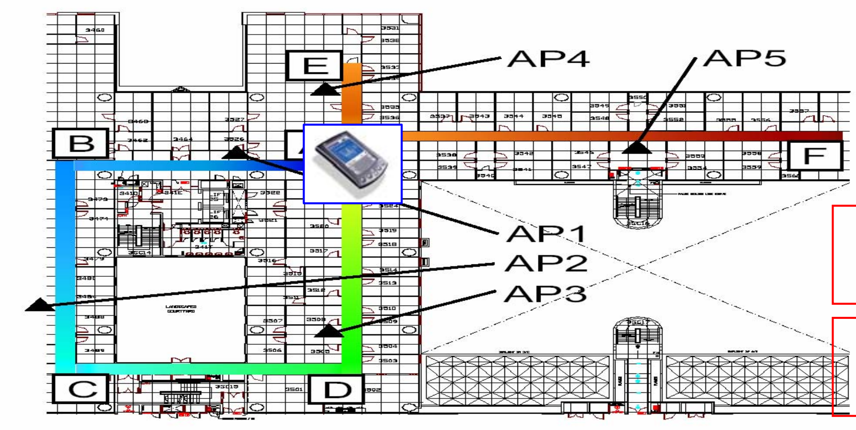
*Propagation-based models use access point locations

*Learning-based models do not use their locations

What Kind of Data We Have?

HKUST

- The Location of Mobile Devices
  - Known when walking by landmarks (corners, doors)
  - Unknown elsewhere
- The Location of Access Points
  - Known for those deployed by us
  - Unknown for those deployed by other persons



Known (x1,y1)

Known (x3,y3)

Unknown (-, -)

Known (x, y)

Unknown (-, -)

Table 1: Signal Strength (unit:dBm)

	AP_1	AP_2	AP_3	AP_4	AP_5
t_A	-40		-60	-40	-70
t_B	-50	-60		-80	
t_C		-40	-70		
t_D	-80		**-40**	-70	
t_E	-40		-70	-40	-80
t_F	-80			-80	-50

(All values are rounded for illustration)

Observation of Signal Strength

HKUST

- A user with a mobile device walks through A B,C,D,E,F

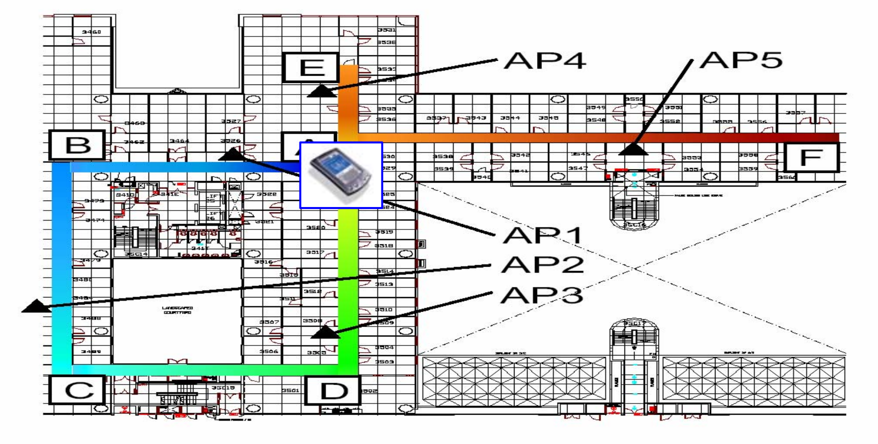


Table 1: Signal Strength (unit:dBm)

	AP_1	AP_2	AP_3	AP_4	AP_5
t_A	-40		-60	-40	-70
t_B	-50	-60		-80	
t_C		-40	-70		
t_D	-80		**-40**	-70	
$t_{A'}$	-40		-70	-40	-60
t_E	-40		-70	-40	-80
t_F	-80			-80	-50

(All values are rounded for illustration)

- Characteristics / Constraints
  - Two **rows** are similar ⇔ Two mobile devices are close (t_A & t_E)
  - Neighbored **rows** are similar ⇔ User Trajectory is smooth (t_i & t_{i+1})
  - Two **columns** are similar ⇔ Two access points are close (AP_1 & AP_4)
  - Strong **cell** ⇔ mobile device and access point are close (t_D at AP_3)

Idea - Latent Semantic Indexing

HKUST

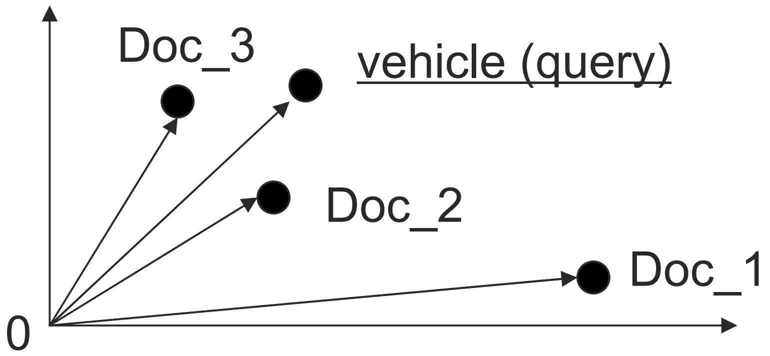
- Term ⇔ Access Point
- Document ⇔ Signal Fingerprint

n Term

m Doc

Doc\Term	moon	car	truck
Doc_1	1	0	0
Doc_2	0	2	1
Doc_3	0	1	2

SVD



n Access Point

m Mobile Device

	AP_1	AP_2	AP_3	AP_4	AP_5
t_A	-40		-60	-40	-70
t_B	-50	-60		-80	
t_C		-40	-70		
t_D	-80		**-40**	-70	
t_E	-40		-70	-40	-80
t_F	-80			-80	-50

SVD

?

Solution of Latent Semantic Indexing

- Transform signal matrix to weight matrix

$$S = [s_{ij}]_{m\times n} \longrightarrow A = [a_{ij}]_{m\times n}$$

- Normalize the weight matrix

$$A_{\mathcal{N}} = D_1^{-1/2} A D_2^{-1/2}.$$

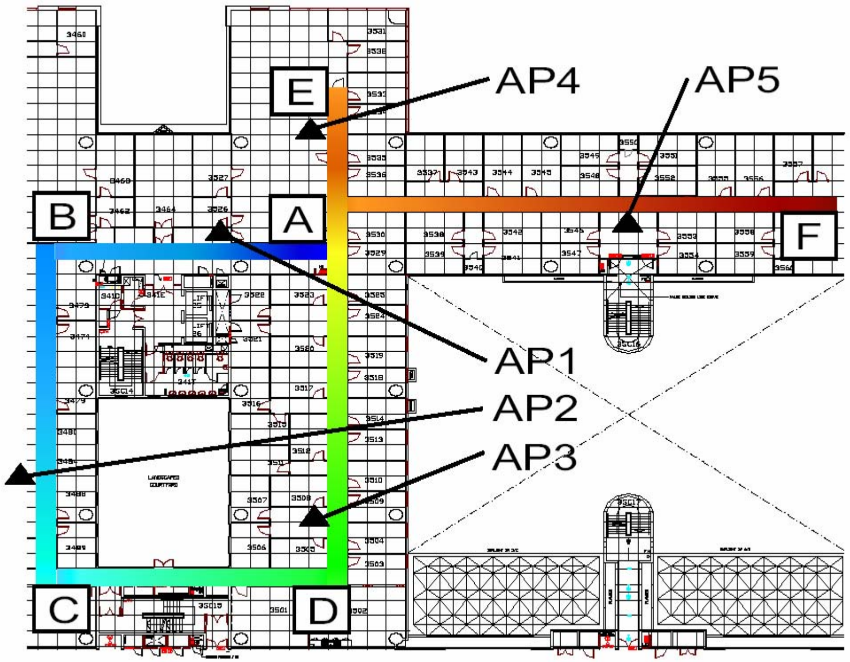
$$D_1 = diag(d_1^1, d_2^1, \ldots, d_m^1) \text{ where } d_i^1 = \textstyle\sum_{j=1}^{n} a_{ij}$$
$$D_2 = diag(d_1^2, d_2^2, \ldots, d_n^2) \text{ where } d_j^2 = \textstyle\sum_{i=1}^{m} a_{ij}.$$

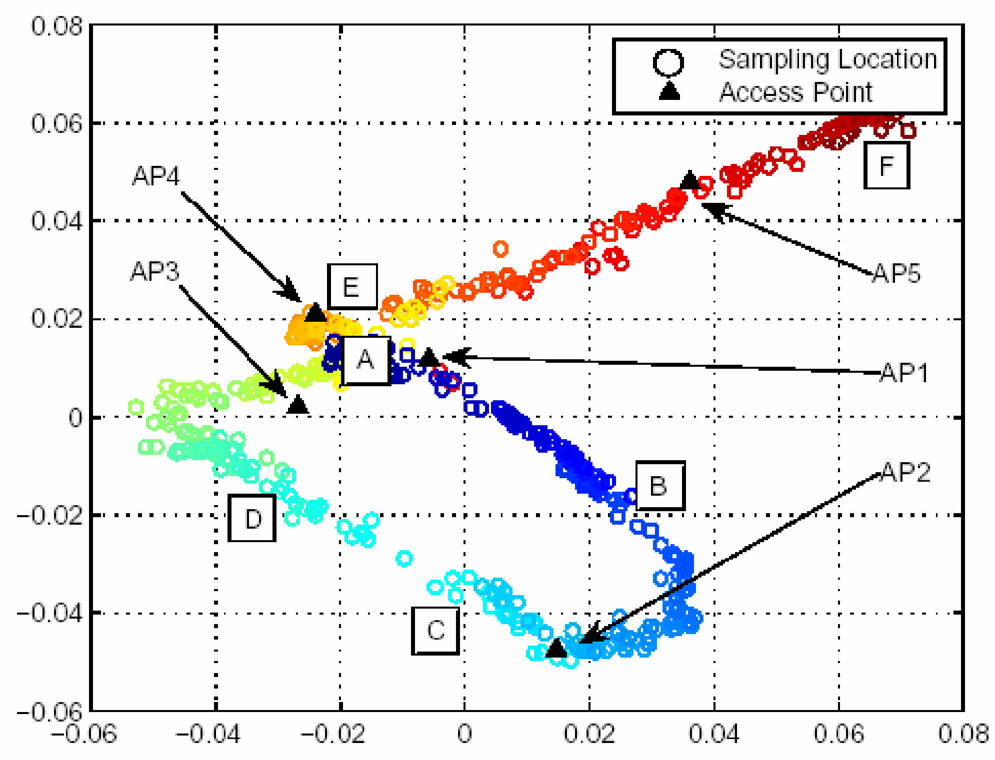
- Recover the relative coordinates by SVD

$$A_{\mathcal{N}} \approx U_{m\times r}\Sigma_{r\times r}V'_{n\times r}.$$

- Notation
  - m mobile devices, n access point, r=2 dimension

Illustration of Latent Semantic Indexing

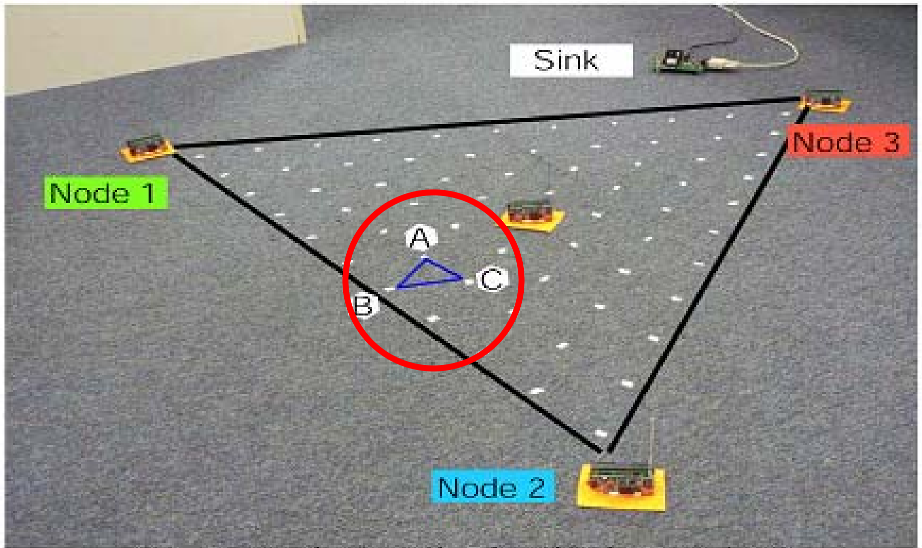




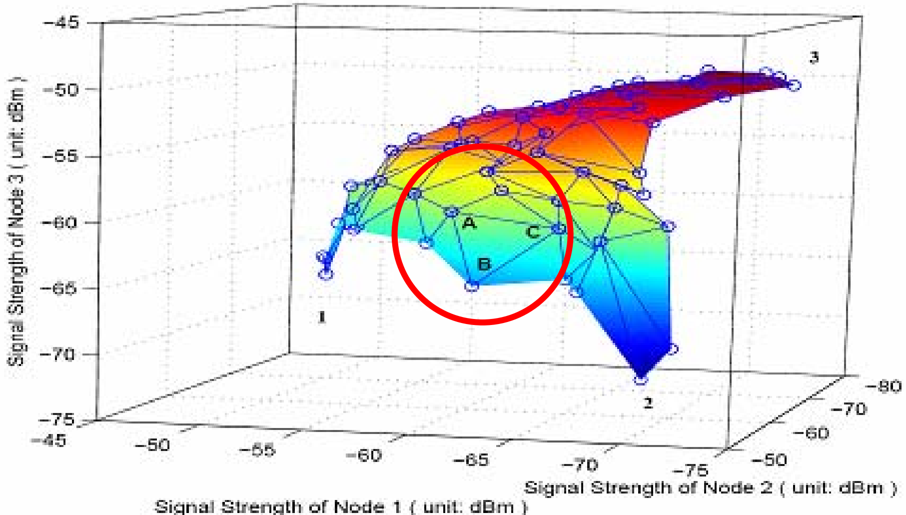
- Retrieve Relative Coordinates / Recover AP locations as well
- Well Alignment between Mobile Device and Access Points

Encode Labels by Manifold Learning

- Neighbors in Locations ⇔ Neighors in Signals ?
- Construct K-Neighborhood Graph for Manifold



(c) Experimental Physical Test-bed



(d) Experimental Signal Manifold

Encode Labels by Manifold Learning

HKUST

- Offline Training Phase (Give labels to unlabeled data)
  - Optimal locations of mobile devices and access points

$$R^* = \arg\min_{R \in \mathbb{R}^{(m+n)\times 2}} \underbrace{(R-Y)'J(R-Y)}_{\text{Encode Labels}} + \underbrace{\gamma R' L R}_{\text{Signal Manifold}}$$

$$R = [\mathbf{r}_1', \mathbf{r}_2, \ldots, \mathbf{r}_{m+n}']' = [P' \; Q']'$$

$$Y = [Y_P' \; Y_Q']' \qquad J = \begin{bmatrix} J_P & 0 \\ 0 & J_Q \end{bmatrix}$$

- Online Localization Phase
  - Use the Property of Harmonic Functions (~Weighted KNN)

$$\tilde{\mathbf{r}}_i \approx \frac{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij} \mathbf{r}_j}{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij}}$$

Encode Labels by Manifold Learning

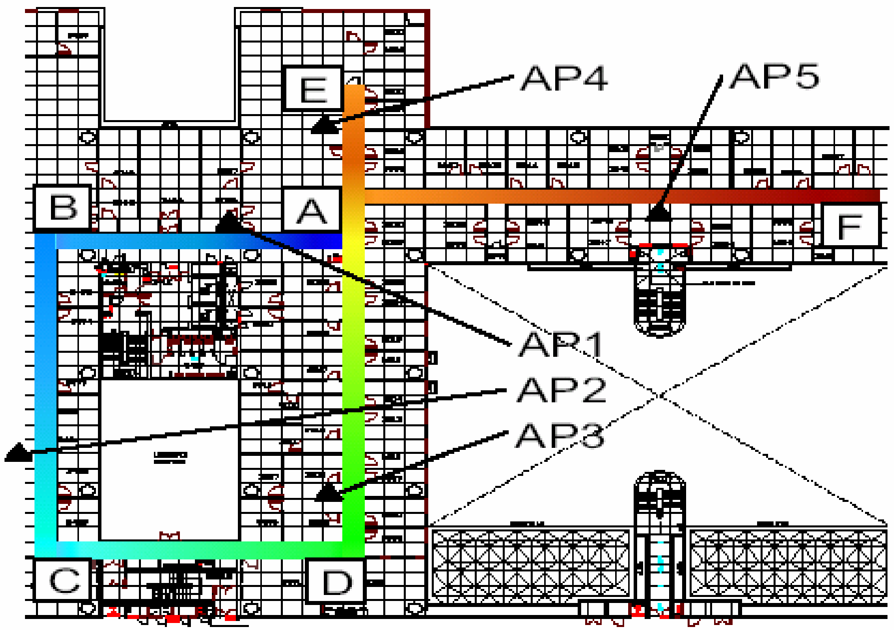
HKUST

$L = D - W$

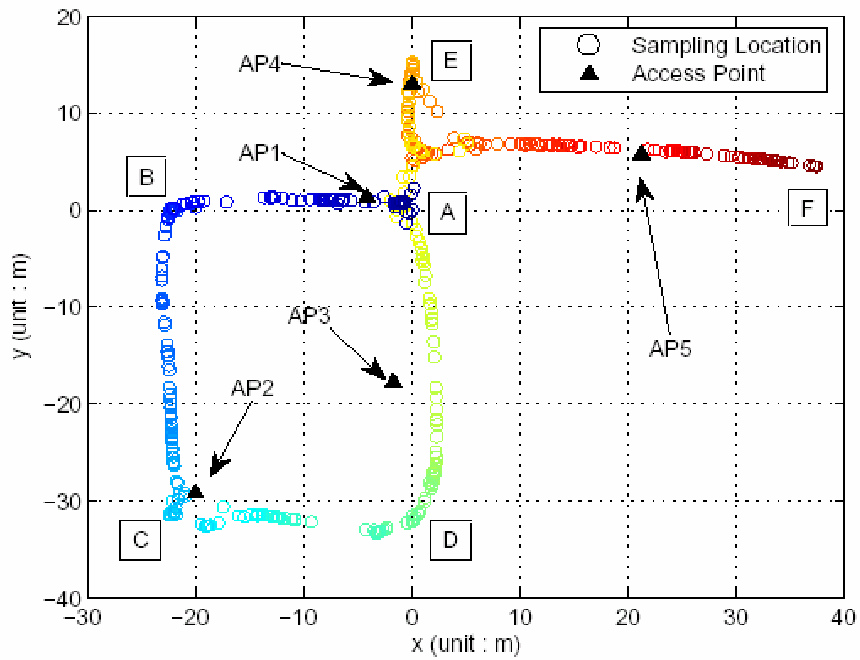
W =

W_P Manifold Matrix Correlation within mobile devices	**$A_{N'}$ Latent Semantic Index** Correlation between mobile devices and access points
A_N Latent Semantic Index Correlation between access points and mobile devices	**L_Q Manifold Matrix** Correlation within access points

Co-Localization Example



802.11 WLAN Test-bed



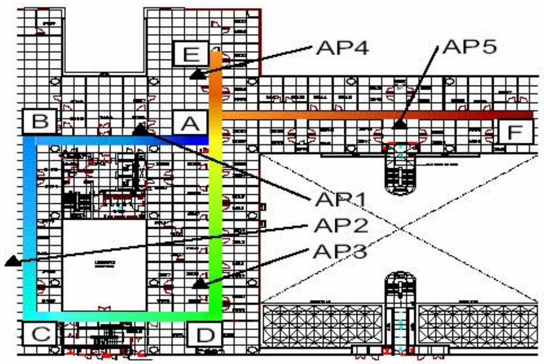
Co-Localization Result

Experimental Setups

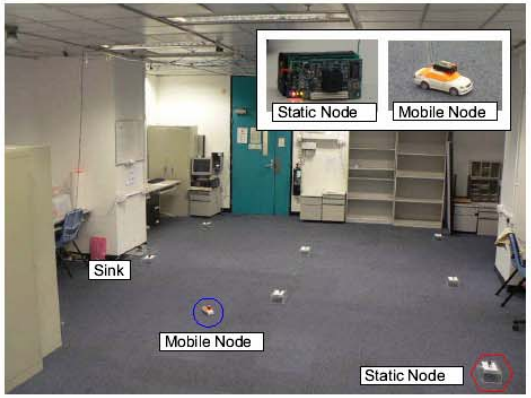
- 802.11 Wireless LAN (WLAN)
- Wireless Sensor Network (WSN)
- Radio-frequency identification (RFID)

Table 2: The experimental setups of WLAN, WSN and RFID

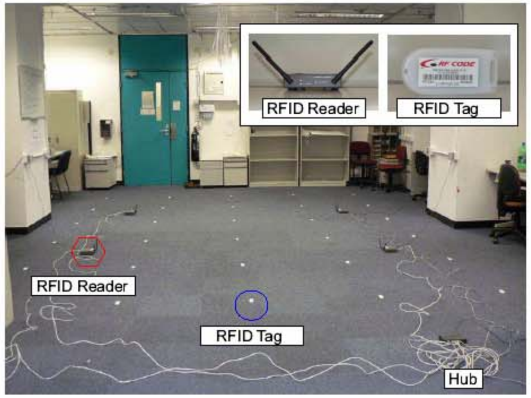
Infrastructure	AP	MD	Test-bed	Scale	Dataset Size	Motion Pattern
WLAN	5 Access Points	1 Notebook	Hallway	$60m \times 50m$	2000	Mobile (robot)
WSN	8 Static Nodes	1 Mobile Node	Room	$5m \times 4m$	4000	Mobile (human)
RFID	4 RFID Readers	30 RFID Tags	Room	$5m \times 4m$	2000	Static



(a) WLAN Test-bed



(b) WSN Test-bed

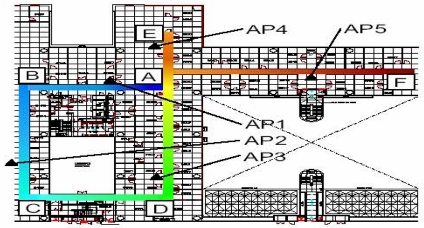


(c) RFID Test-bed

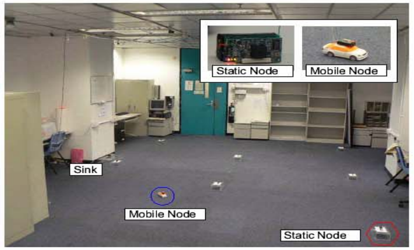
Tests on WLAN / WSN / RFID

Different

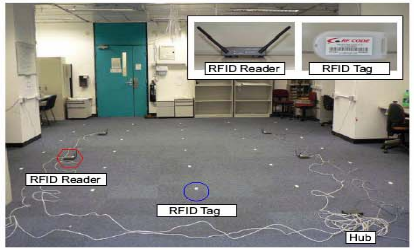
Test-beds



(a) WLAN Test-bed



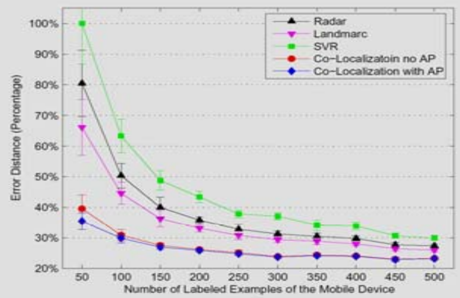
(b) WSN Test-bed



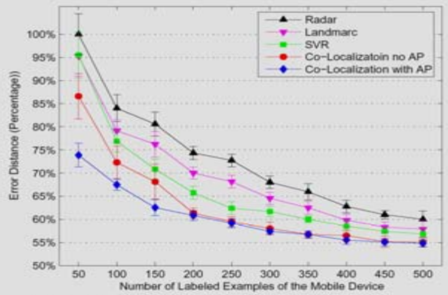
(c) RFID Test-bed

Locate

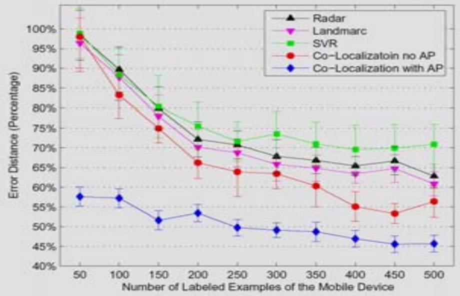
Mobile Devices



(c) WLAN MD (notebook)



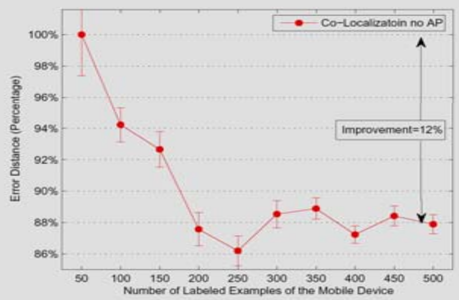
(b) WSN MD (mobile sensor node)



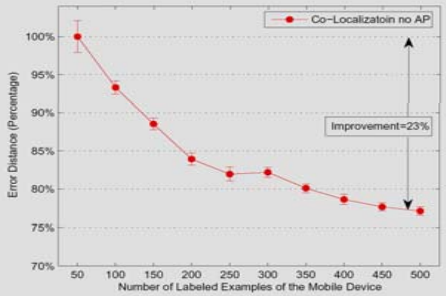
(a) RFID MD (tags)

Locate

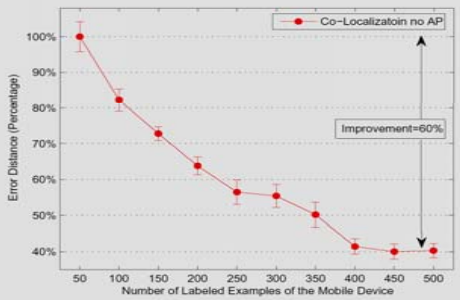
Access Points



(f) WLAN AP (access points)



(e) WSN AP (static sensor nodes)



(d) RFID AP (readers)

Question Four

HKUST

Can we update the model online?

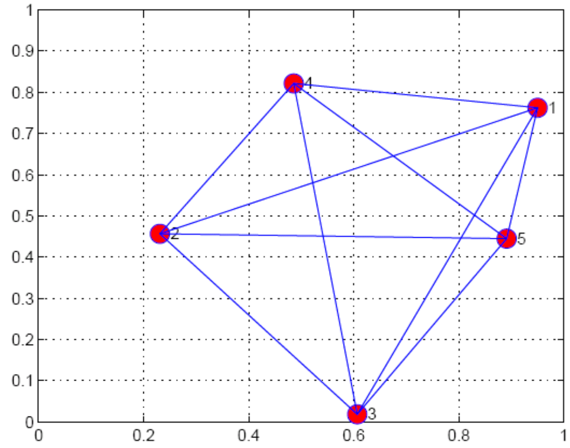
- The previous models are operated in a traditional offline/online manner
- How to adapt new data without re-training everything?

Online Co-Localization

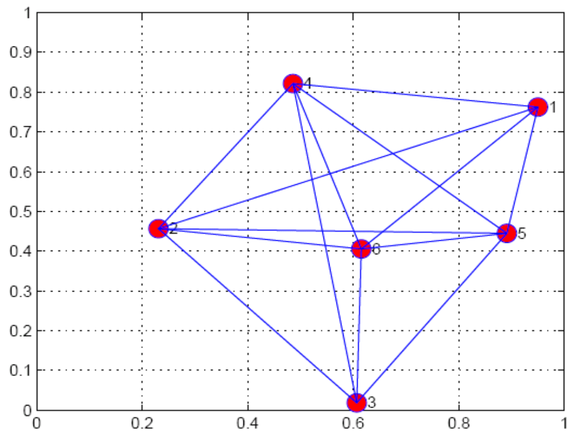
HKUST

- Predict Step
  - Use Weighted KNN as initial estimation of the Coordinate
- Update Step
  - Update the K-Neighborhood in Manifold
  - Update Coordinate Iteratively by

$$\mathbf{r}_i^{new} \leftarrow \frac{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij} \mathbf{r}_j^{old}}{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij}} \quad (i = 1, 2, \ldots, m+n)$$



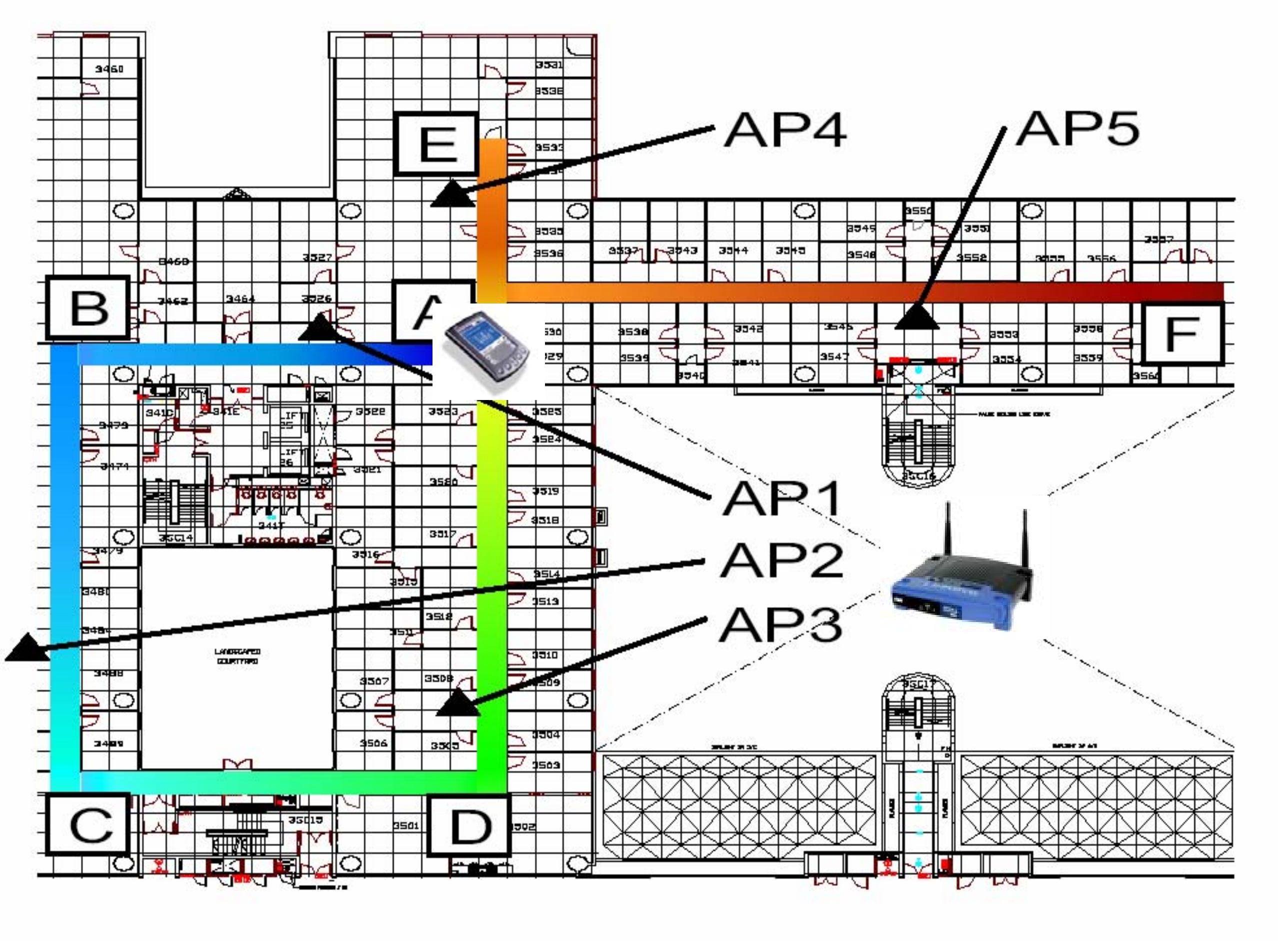
(a) A 4-Nearest-Neighbor graph that has 5 nodes



(b) Add Node 6 to the graph

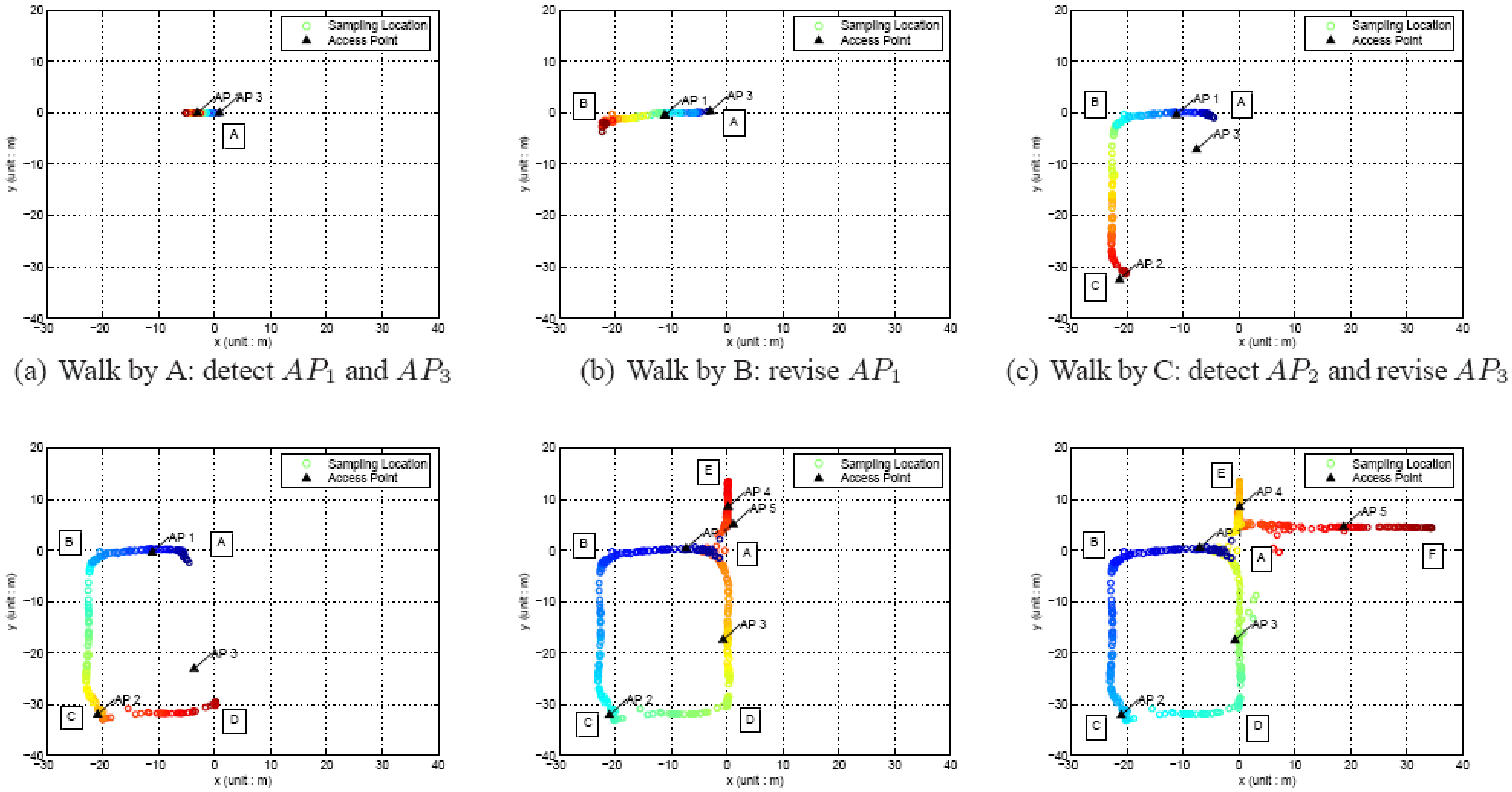
Experimental Setups

- 802.11 Wireless LAN (WLAN)



Online Co-Localization [movie]

HKUST



(a) Walk by A: detect AP_1 and AP_3 (b) Walk by B: revise AP_1 (c) Walk by C: detect AP_2 and revise AP_3

(d) Walk by D: revise AP_3 (e) Walk by E: detect AP_4 and AP_5 (f) Walk by F: revise AP_5

Figure 2: Illustration of the Online Co-Localization when a user walks from A through $B, \ldots, D$ to F

Model Update Speed

- 10 times faster than its two-phase counterpart
- Accuracy is the same as the two-phase method

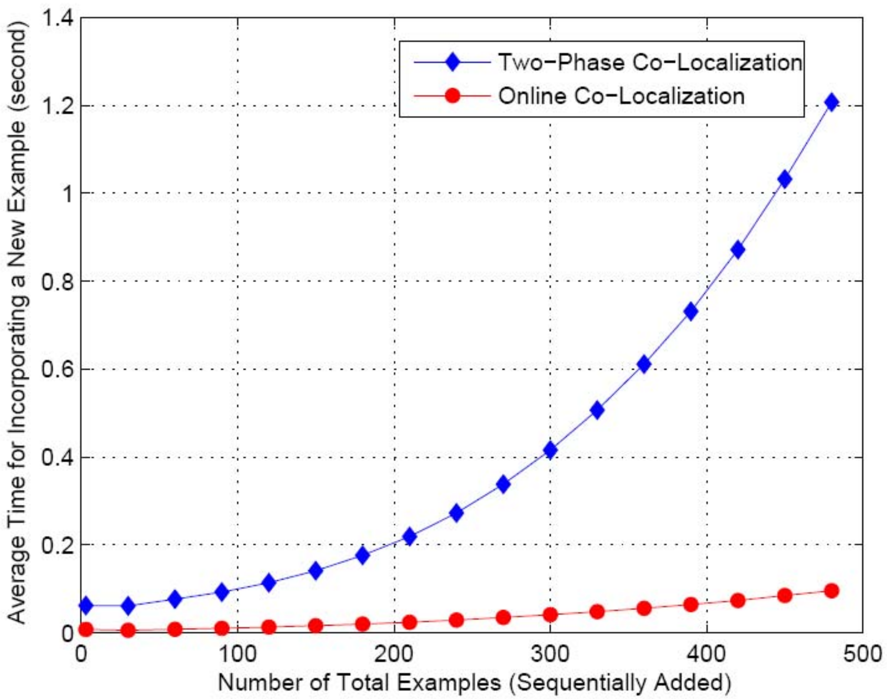
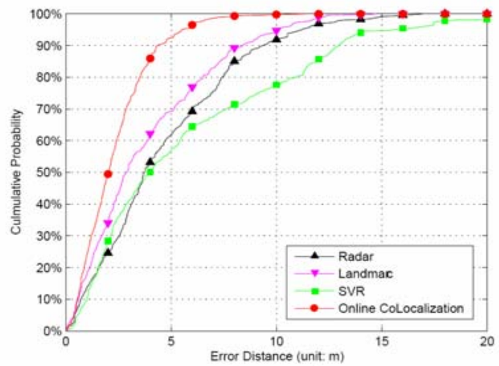
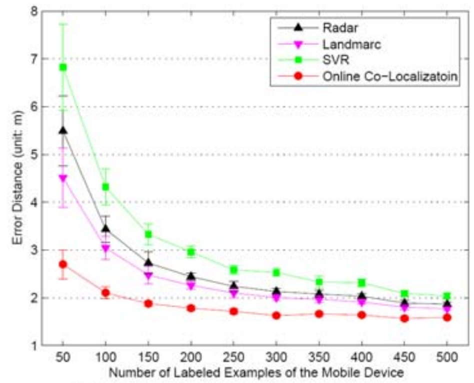


Figure 4: Average Running Time Comparison



(a) Comparison of Accuracy



(b) WLAN Mobile Devices

Summary

HKUST

- Radio-Signal-Strength-based Tracking
  - RSS-based Tracking
  - Application Scenario
  - Radio Characteristics
  - Localization Models
- Increase Accuracy
  - LeKCCA (IJCAI-2005)
- Reduce Calibration
  - LeMan (AAAI-2006)
- Encode Further Information Sources
  - Co-Localization (IJCAI 2007)
- Update Model Online
  - Online Co-Localization (AAAI-2007)
- Conclusion

Development of Our Models

HKUST

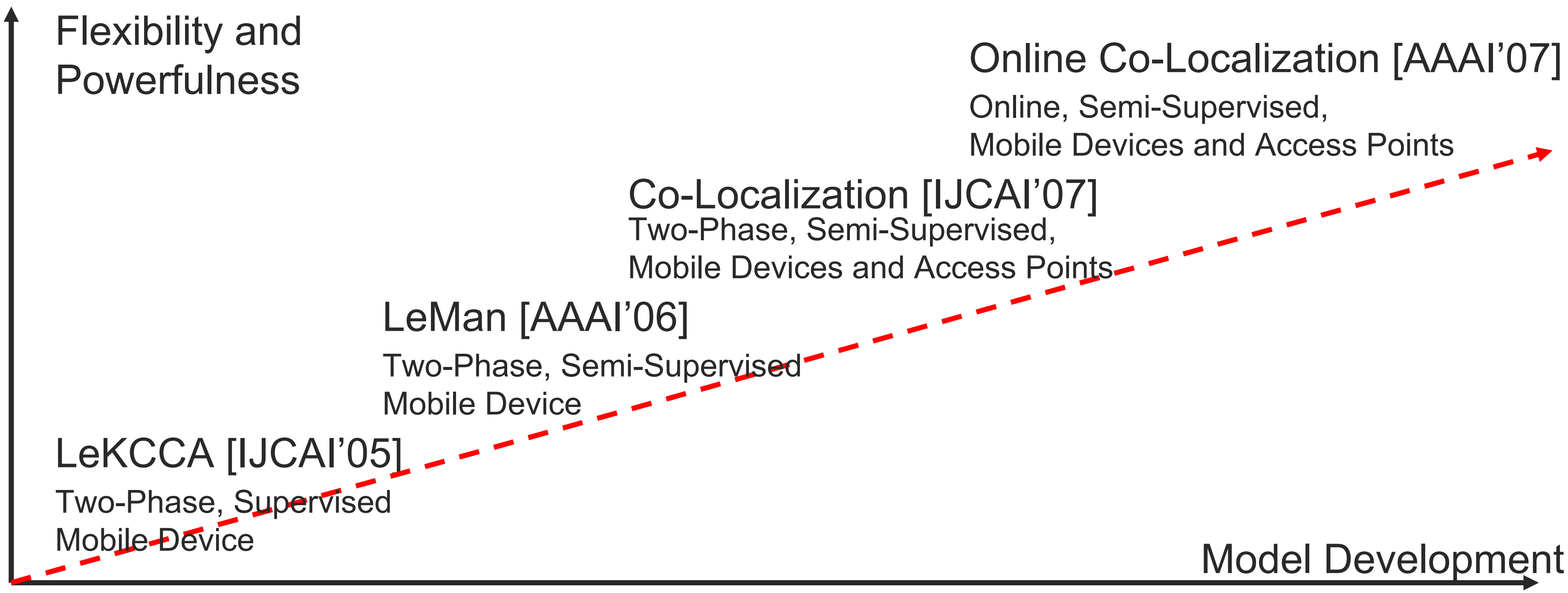


Table 10.1: Characteristics of Our Models

Method / Calibration	Access Point		Mobile Device		Trace	Online
	Unlabeled	Labeled	Unlabeled	Labeled		
LeKCCA	✓			✓		
LeMan	✓		✓	✓		
Co-Localization	✓	✓	✓	✓		
Online Co-Localization	✓	✓	✓	✓	✓	✓

The End

HKUST

Thank You

Question ?