Indoor Positioning with WiFi Signals

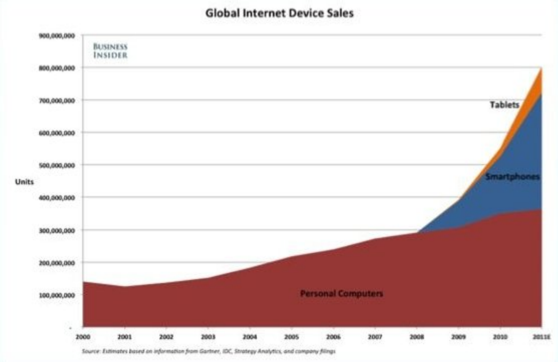
Device Self-Calibration using Histograms

Christos Laoudias*, Robert Piché† and Christos Panayiotou*

*KIOS Research Center for Intelligent Systems and Networks, University of Cyprus
†Tampere University of Technology, Tampere, Finland

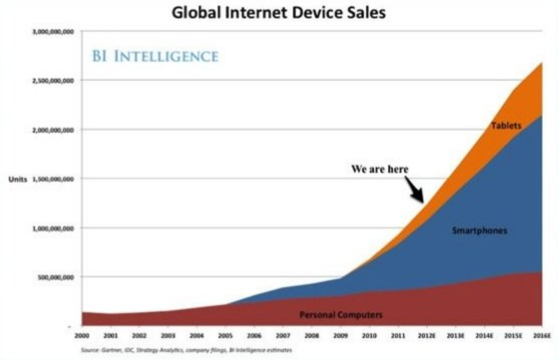
Inside or Outside?

Smartphone Facts



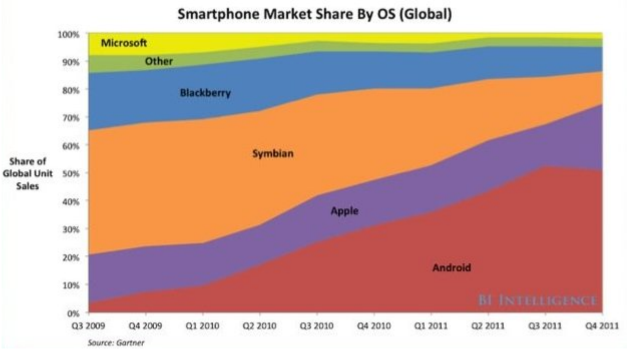
- In 2011 the smartphone sales outnumbered the PC sales for the first time

Smartphone Facts



- In 2011 the smartphone sales outnumbered the PC sales for the first time
- Predictions suggest that smartphones will dominate the computing device market

Smartphone Facts



- In 2011 the smartphone sales outnumbered the PC sales for the first time
- Predictions suggest that smartphones will dominate the computing device market
- The Android is now (and will probably remain) the leading OS

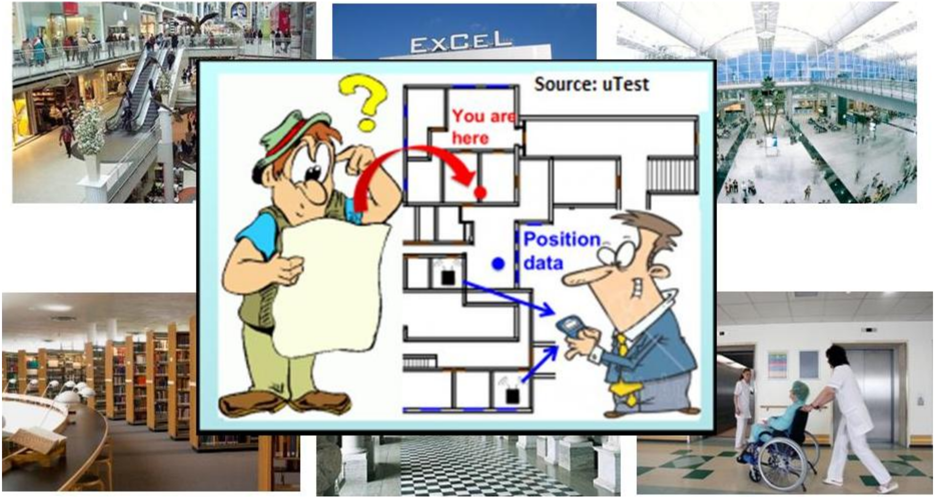
Target Indoor Environments





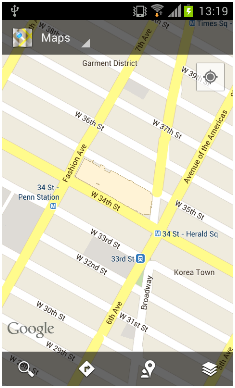
Source: google images

Target Indoor Environments

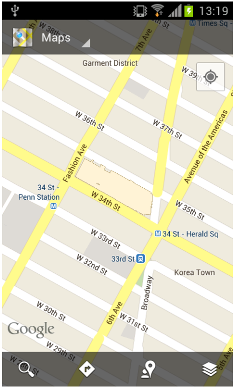


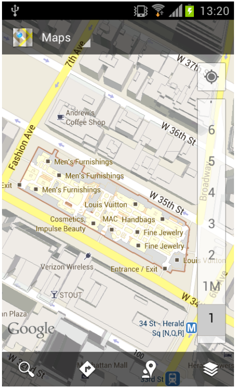
Source: google images

Google Maps Mobile Goes Indoor

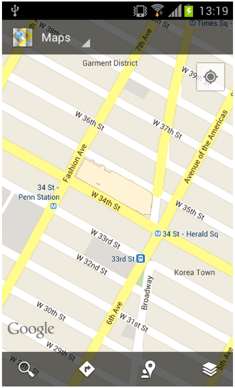


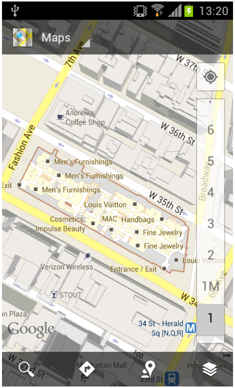
Google Maps Mobile Goes Indoor

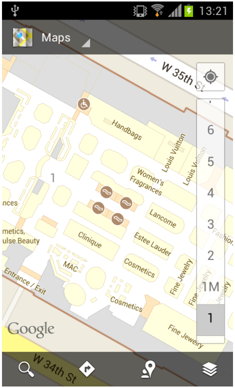




Google Maps Mobile Goes Indoor

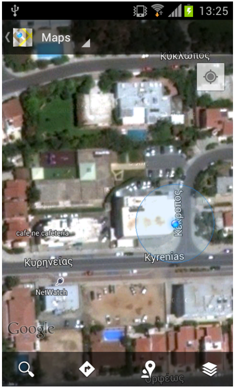




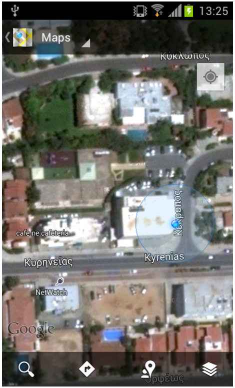


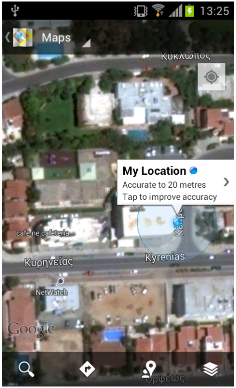
- Launched in 2011 with 60 venues in the U.S. and 50 in Japan
- Now has ∼10,000 maps and offers indoor walking directions
- Indoor geo-location based on WiFi signal triangulation (rumoured)

What about accuracy?



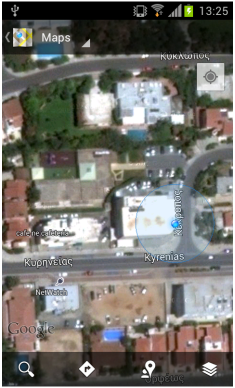
What about accuracy?

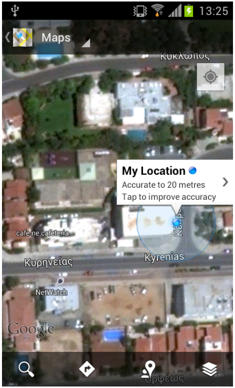


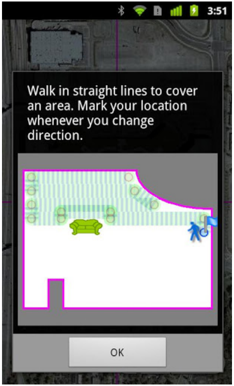


- The accuracy is around 20–50m (depending on the AP density)

What about accuracy?



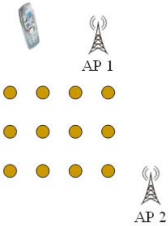




- The accuracy is around 20–50m (depending on the AP density)
- Google Floor Plan Marker app promises to improve accuracy (floorplan map is required)

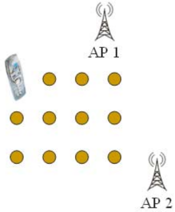
Signal Strength Fingerprints

Fingerprint-based Positioning



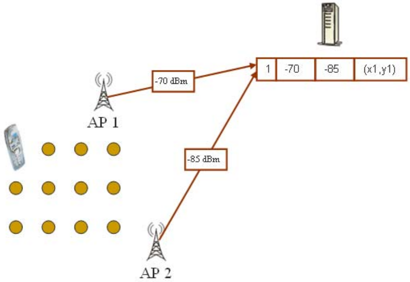
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M}\sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i,\ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



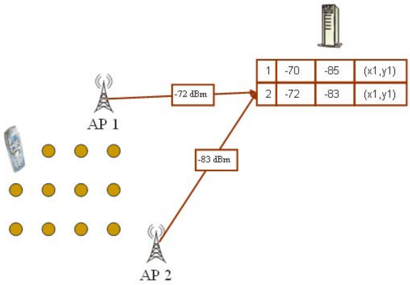
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \;\; i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



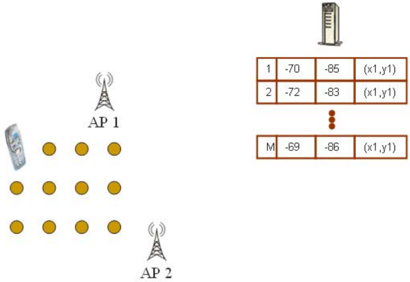
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \dots, r_{in}]^T$
  - Averaging $\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i,\ i = 0, \dots, N_d$
  - New fingerprint $s = [s_1, \dots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



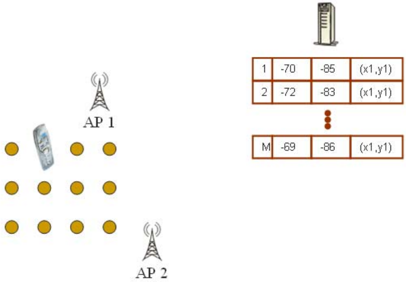
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \dots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \;\; i = 0, \dots, N_d$
  - New fingerprint $s = [s_1, \dots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



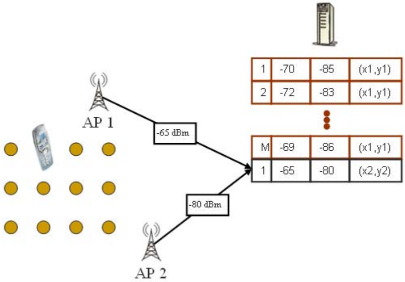
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i,\ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



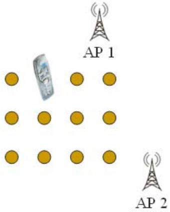
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \; i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

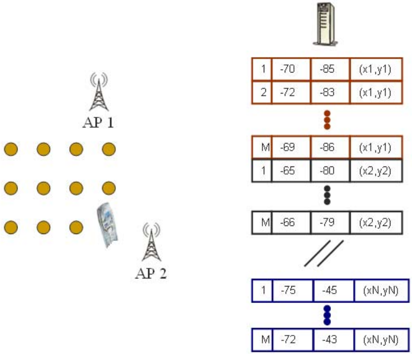
Fingerprint-based Positioning



1	-70	-85	(x1,y1)
2	-72	-83	(x1,y1)
⋮			
M	-69	-86	(x1,y1)
1	-65	-80	(x2,y2)
⋮			
M	-66	-79	(x2,y2)

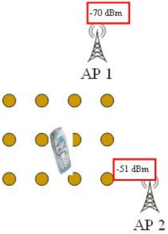
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i,\ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \; i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

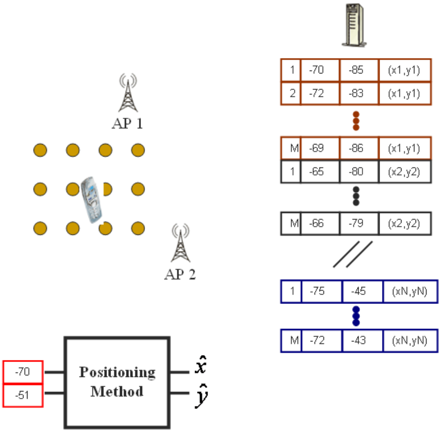
Fingerprint-based Positioning



1	-70	-85	(x1,y1)
2	-72	-83	(x1,y1)
M	-69	-86	(x1,y1)
1	-65	-80	(x2,y2)
M	-66	-79	(x2,y2)
1	-75	-45	(xN,yN)
M	-72	-43	(xN,yN)

- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\overline{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Fingerprint-based Positioning



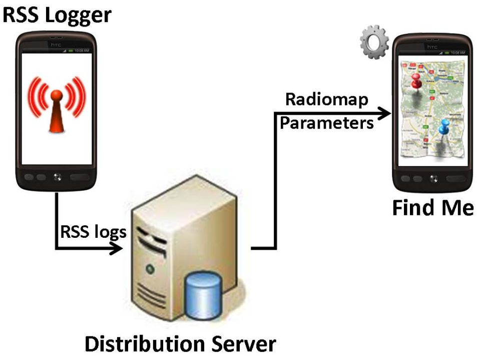
- **Offline phase:** Build RSS radiomap with a reference device D_0
  - n APs deployed
  - Fingerprints $r_i = [r_{i1}, \ldots, r_{in}]^T$
  - Averaging $\bar{r}_i = \frac{1}{M} \sum_{m=1}^{M} r_i(m)$
- **Online phase:** Positioning with device $D_i, \ i = 0, \ldots, N_d$
  - New fingerprint $s = [s_1, \ldots, s_n]^T$
  - Calculate $\widehat{\ell}$ using the radiomap

Airplace: Indoor Positioning on Android Devices

Terminal-based Infrastructure-assisted Architecture

- **Low Communication Overhead**: Avoids uploading the observed RSS fingerprint to the positioning server
- **User Privacy & Security**: Location is estimated by the user

http://www2.ucy.ac.cy/~laoudias/pages/platform.html



Device Diversity

- RSS is intended for determining the signal quality and not for positioning purposes

Device Diversity

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - The implementation of each vendor is limited in [0 $\mathrm{RSSI}_{\mathrm{max}}$]
  - RSSI is mapped to actual power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging material

Device Diversity

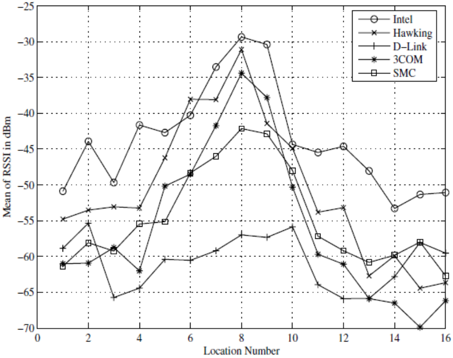
- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - The implementation of each vendor is limited in [0 $\mathrm{RSSI}_{\mathrm{max}}$]
  - RSSI is mapped to actual power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging material
- Using a new device for positioning is feasible, but the RSS values are not compatible with the radiomap, leading to accuracy degradation

Device Diversity

- RSS is intended for determining the signal quality and not for positioning purposes
- Different devices do not report RSS values in the same way
  - The WiFi standard (IEEE 802.11) defines the RSS Indicator (1 byte integer) for measuring RSS in [0 255]
  - The implementation of each vendor is limited in [0 $\mathrm{RSSI}_{\max}$]
  - RSSI is mapped to actual power values in dBm internally by the device driver (proprietary information)
  - Even worse: same chipsets may not report the same RSS values due to different antennas or packaging material
- Using a new device for positioning is feasible, but the RSS values are not compatible with the radiomap, leading to accuracy degradation
- Best accuracy is guaranteed only if the the user carries the same device during positioning, otherwise a *calibration* step is required

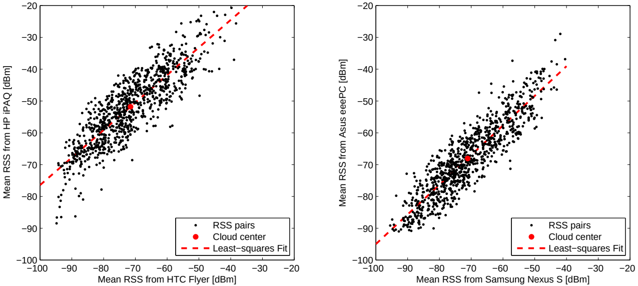
Device Diversity

Vendor	Model	Chipset	Max (dBm)	Min (dBm)	Range
3COM	3CRUSB10075	unknown	+10	-94	104
D-Link	AirPlus DWL-650+	Texas Instrument	-50	-100	50
SMC	EZ Connect SMC2635W	ADMTek	-14	-82	68
Hawking Technology	HWC54G Rev.R	Prism GT	0	-75	75
Intel	PRO/Wireless 2200BG	Intel	-10	-84	74



Source: K. Kaemarungsi (2006)

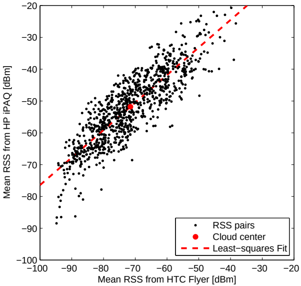
Good News: Linear relation between RSS values

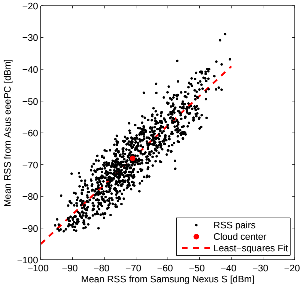


- **Manual Calibration**: Collect several colocated RSS pairs at *known* locations and estimate the linear coefficients through least squares

$$\bar{r}_{ij}^{(2)} = \alpha_{12}\bar{r}_{ij}^{(1)} + \beta_{12}$$

Good News: Linear relation between RSS values





- **Manual Calibration**: Collect several colocated RSS pairs at *known* locations and estimate the linear coefficients through least squares

$$\bar{r}_{ij}^{(2)} = \alpha_{12}\bar{r}_{ij}^{(1)} + \beta_{12}$$

- Limited Applicability: (i) User needs to be familiar with the indoor area and (ii) a considerable data collection effort is required

Can we do it more efficiently?

Objectives

- Fully automatic approach with short calibration time
- Runs concurrently with positioning while the user walks around
- No user intervention or tedious data collection

Idea

- Perform device self-calibration on-the-fly using histograms of RSS values observed simultaneously with positioning

RSS Histograms

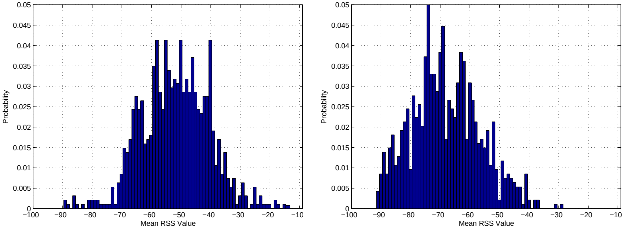


Figure: HP iPAQ (left) and Asus eeePC (right)

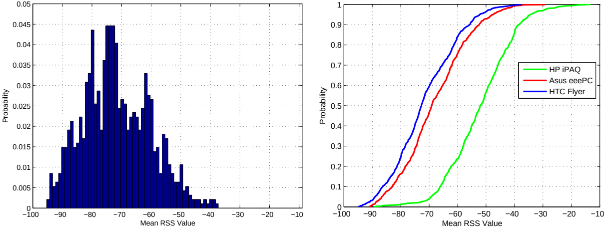
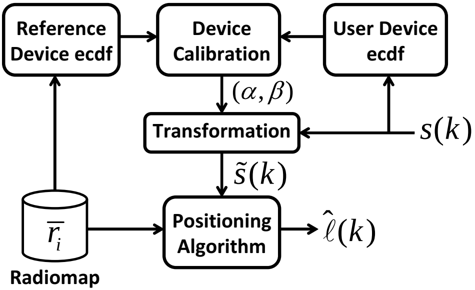


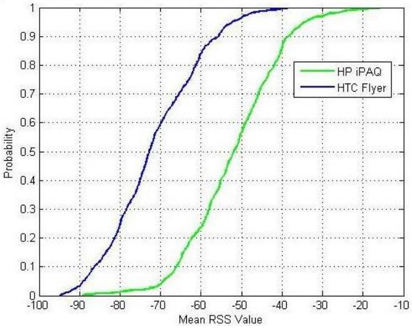
Figure: HTC Flyer (left) and Empirical cdfs (right)

Self-Calibration Method



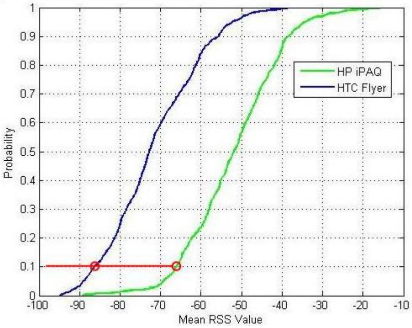
1. Create the RSS ecdf of the reference device by using the radiomap
2. Create and update the ecdf of the new device by using $s(k)$
3. Fit a linear mapping between the reference and new device to obtain the parameters (α, β) by using "representative" ecdf values
4. Transform the observed RSS values with $\tilde{s}_j(k) = \alpha s_j(k) + \beta$
5. Estimate location $\hat{\ell}(k)$ with any fingerprint-based algorithm

Inverse ecdf Linear fitting



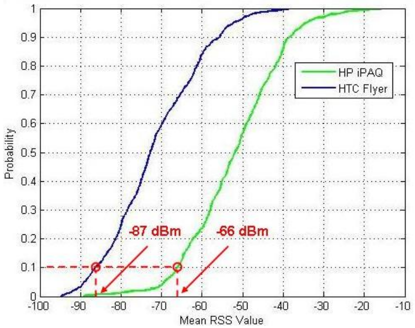
- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ denote the ecdf of the reference and user device
- Obtain (α, β) from $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta, \; y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Inverse ecdf Linear fitting



- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ denote the ecdf of the reference and user device
- Obtain (α, β) from $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta, \; y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Inverse ecdf Linear fitting



- $F(x)$ gives the probability that the RSS value is less than x, $F^{-1}(y)$ returns the RSS value that corresponds to the y-th cdf percentile
- $F_r(x)$ and $F_u(x)$ denote the ecdf of the reference and user device
- Obtain (α, β) from $F_r^{-1}(y) = \alpha F_u^{-1}(y) + \beta, \;\; y \in \{0.1, 0.2, \ldots, 0.9\}$
- (α, β) are initialized to $(1, 0)$ and updated periodically (e.g. every 10 sec) thereafter, while the user is walking

Experimental Results

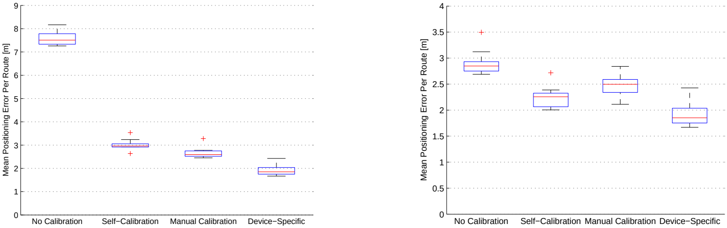


Figure: HTC Flyer user with HP iPAQ (left) or Asus eeePC (right) radiomap

Experimental Results

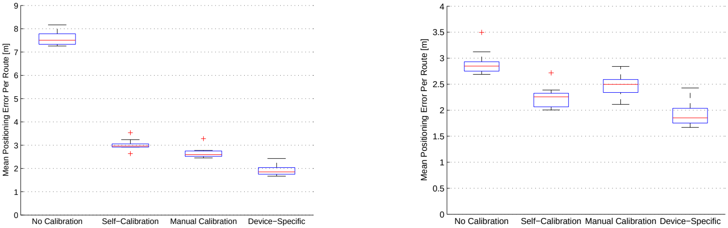
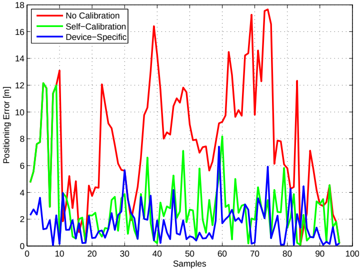


Figure: HTC Flyer user with HP iPAQ (left) or Asus eeePC (right) radiomap



Results with 5 devices

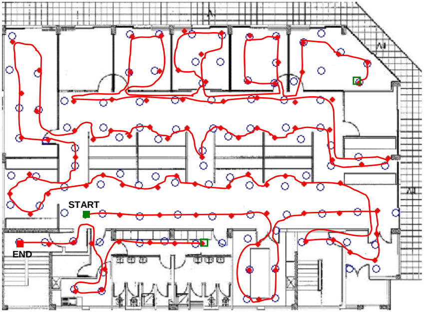
Table: Median of the mean error $\bar{\epsilon}$ [m], with and without calibration.

	iPAQ	**eeePC**	**Flyer**	**Desire**	**Nexus S**
iPAQ	2.7	2.8 (6.6)	3.0 (7.5)	2.9 (8.4)	2.6 (7.7)
eeePC	2.8 (4.4)	2.3	2.3 (2.8)	2.6 (3.5)	2.5 (2.9)
Flyer	3.2 (5.9)	2.6 (3.0)	1.9	2.1 (2.3)	2.6 (2.7)
Desire	3.4 (6.1)	2.8 (3.2)	2.5 (2.5)	2.4	2.5 (2.6)
Nexus S	3.0 (6.2)	2.6 (2.8)	2.7 (2.7)	2.4 (2.5)	2.3

Thank you for your attention
Questions?

Extra Slides

Experimental Setup @ KIOS



- 560 m^2 office, 9 WiFi APs, 5 devices (1 HP iPAQ PDA, 1 Asus eeePC laptop, 1 HTC Flyer Android tablet, 2 Android smartphones)
- **Training Data:** 105 reference locations, 20 fingerprints per location (2100 in total) with each device for comparison
- **Testing Data:** Route with 2 segments, 96 test locations, 1 fingerprint per location, route sampled 10 times

Inverse ecdf Least Squares Fitting

If $\mathbf{u}$ is a continuous random variable and $\mathbf{y} = f(\mathbf{u})$ with monotonically increasing f then $f = F_{\mathbf{y}}^{-1} \circ F_{\mathbf{u}}$. In particular, the inverse cdf ordered pairs

$$\{(u_i, y_i) = (F_{\mathbf{u}}^{-1}(q_i), F_{\mathbf{y}}^{-1}(q_i)) : q_i \in \{0.1, \ldots, 0.9\}\}$$

lie on the curve $y = f(u)$.

Proof:
We have

$$\begin{aligned} F_{\mathbf{u}}(u) &= P(\mathbf{u} \le u) = P(f(\mathbf{u}) \le f(u)) = \\ &= P(\mathbf{y} \le f(u)) = F_{\mathbf{y}}(f(u)). \end{aligned}$$

Applying $F_{\mathbf{y}}^{-1}$ to both sides gives the identity $f = F_{\mathbf{y}}^{-1} \circ F_{\mathbf{u}}$. Also, the components of the inverse cdf ordered pairs satisfy

$$y_i = F_{\mathbf{y}}^{-1}(q_i) = F_{\mathbf{y}}^{-1}(F_{\mathbf{u}}(u_i)) = f(u_i).$$