Fast and Accurate Indoor Localization

Duc A. Tran

Director, Network Information Systems Lab
Department of Computer Science
University of Massachusetts, Boston

Indoor Localization: Demand

PRWeb
Online Visibility from Vocus

HOME NEWS CENTER BLOG

Front Page | Arts | Business | Education | Environment | Government | Indu

Thursday, July 25, 2013

Global Indoor Location, Indoor Positioning and Indoor Navigation (IPIN) Market to Grow Up to $2.60 Billion in 2018 - New Report by MarketsandMarkets

Apple Acquires Indoor GPS Startup WiFiSlam For $20M

RIP EMPSON

Sunday, March 24th, 2013 — 48 Comme

The Wall Street Journal is reporting today that Ap has acquired WiFiSlam, an indoor GPS startup that enables a smartphone to pinpoint its location — alon with that of your friends — in realtime up to 2.5 mete in accuracy.

Apple paid $20 million to acquire WiFiSlam, althougl the specific terms of the deal have not been shared of yet. However, Apple has confirmed the acquisition

Indoor Localization: Requirement

Fast?

"Where am I?" – "Let me think!"

- Think "critical"! (fire rescue, evacuation in a building)
- Think "scalable"! (service to thousands of shoppers in a mall)

Accurate?

"Where am I" – "Apple Store, but I am only 50-50 sure"

(source: studentnewsdaily.com)

Outline

Review of representative indoor localization approaches

- Most focus on accuracy but not computational efficiency

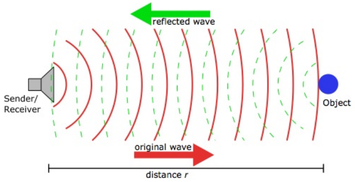
Our recent research results

- Faster positioning based on spatially hierarchical learning
- Better accuracy based on graph regularization
- Truly calibration-free online localization

Ranging

Ranging is a well-established method

- SONAR (= SOund Navigation And Ranging),
- RADAR (= RAdio Detection And Ranging) and
- LIDAR (= LIght Detection And Ranging).



Distance between a transmitter and a receiver can be estimated by probe measurements of signal strength, time of arrival, etc.

Location is determined by multi-lateration/angulation, given distances, or a combination of distances and angles, from a set of references.

Fingerprinting

- Ranging-free (ranging is expensive!)
- Fingerprint = a vector of features $\mathbf{x} = (x_1, x_2, ..., x_m)$ associated with a location
  - should be location-discriminative, easy to obtain
  - e.g,. RSS from Wi-Fi APs, FM towers, sound, light, earth's magnetic

1. Survey: construct a fingerprint map $\{fingerprint\ \mathbf{x}_i,\ location\ \mathbf{y}_i\}_{i=1}^{n}$
2. Training: learn a prediction function: $f(fingerprint\ \mathbf{x}) = location\ \mathbf{y}$
   - Nearest Neighbors (kNN)
   - Bayesian inference
   - Support Vector Machines (SVM), Artificial Neural Networks (ANN)
3. Positioning: for each new fingerprint $\mathbf{x}_{new}$, output $f(\mathbf{x}_{new})$.

Fingerprinting vs. Ranging

- Fingerprinting is preferable for its simplicity, wide applicability, and cost, more suitable for indoor location-based services, especially targeting consumers
- Ranging is expensive, more suitable for specialized highly-critical applications

This talk is focused on the fingerprinting approach

Bayesian Method

$$P(location \mid fingerprint) = \frac{P(fingerprint \mid location) \times P(location)}{P(fingerprint)}$$

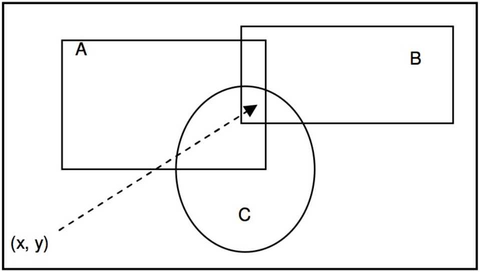
<u>Location estimate for fingerprint **x**</u> is

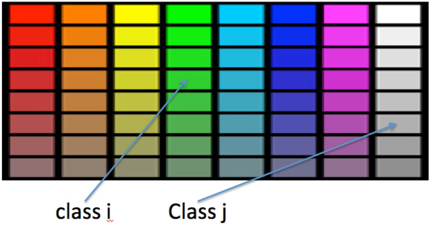
$$f(\mathbf{x}) = \underbrace{\text{argmax}}_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x})$$

<u>Need to know</u> (from training data)

- Location distribution
- Fingerprint distribution at each location
- Fingerprint distribution across the whole area

Classification Method





1. Area = a right set of classes each representing a geographic
   - Usually, the Grid Approach: a class = cell of flat grid
2. Location $\leftarrow$ fingerprint's memberships in classes
   - Support Vector Machines (SVM): de facto for classification

Positioning phase: time complexity $= \mathcal{O}(\textit{no.classes}) = \mathcal{O}(\textit{area})$

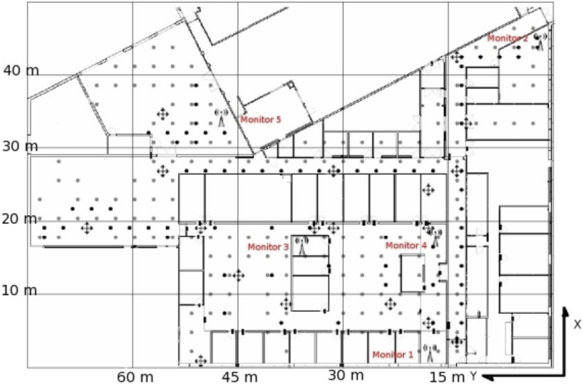
Our Approach: Hierarchical Classification Method

Hierarchical vs. Grid = Binary Search vs. Exhaustive Search

- For each dimension, define $class_i$ = i^{th} percentile of the area
- Consequently, $class_1 \subset class_2 \subset ... \subset class_{100}$
- Find smallest $class_i$ containing fingerprint $\Rightarrow$ location = $i + \frac{1}{2}$

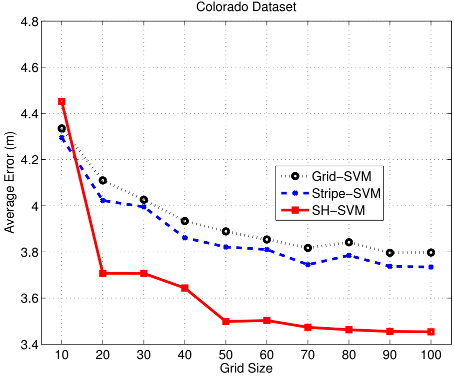
No. classes = $\mathcal{O}(\sqrt{area})$
Positioning phase: time $\mathcal{O}(\log(no.classes)) = \mathcal{O}(\log(\sqrt{area})$

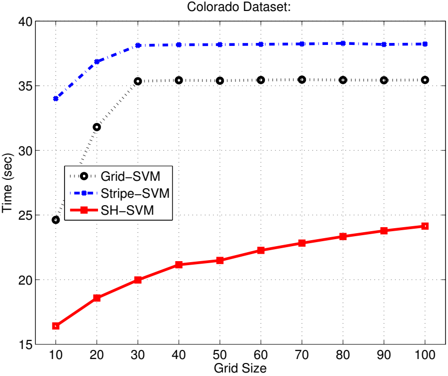
Numerical Results: Wi-Fi Colorado Dataset



- Wi-Fi RSSI fingerprints at 179 sample locations, 5 Wi-Fi APs
- Training set: 1,576 fingerprints (8.8 fingerprints per location)
- Testing set: 77,516 fingerprints.

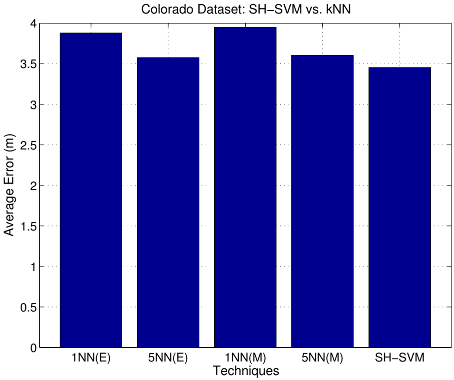
Numerical Results: Hierarchical vs. Grid

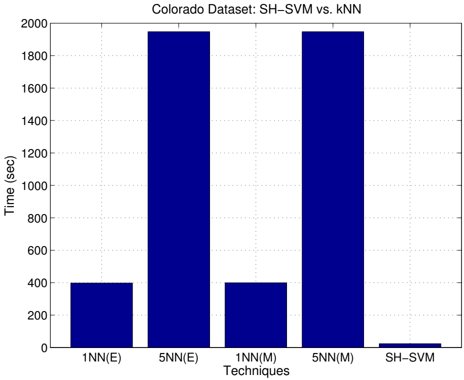




Location error: slightly better
Computation time: 30+% faster

Numerical Results: Hierarchical vs. kNN





Location error: slightly better
Computation time: 16-76 times faster

Enriching Training Set

Fingerprinting is accurate only if training data is sufficiently rich

- Survey phase is labor-intensive!
- Need repeated surveys to adapt to changes in the environment

Why not utilize “unlabeled" samples?

- Fingerprints are abundant if location label is not required.
- Semi-supervised learning to propagate the labels for the unlabeled fingerprints
- Augmented training set $\Rightarrow$ better localization accuracy!

Incomplete Training Set

Input: an incomplete training set

$$\underbrace{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_l, y_l)}_{labeled}, \underbrace{(\mathbf{x}_{l+1}, N/A), ..., (\mathbf{x}_n, N/A)}_{unlabeled}.$$

Output: a good complete training set

$$\underbrace{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_l, y_l)}_{labeled}, \underbrace{(\mathbf{x}_{l+1}, y_{l+1}), ..., (\mathbf{x}_n, y_n)}_{unlabeled}.$$

Location predictor: function $f \in \mathcal{H}_K$, a reproducing kernel Hilbert space (RKHS) with a positive definite kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

$$\min_f J(f) = \frac{1}{l} \sum_{i=1}^{l} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_K^2 + \gamma S(f), \tag{1}$$

where

- 1st term: loss function
- 2nd term: smoothness with respect to kernel K.
- 3rd term: smoothness with respect to an intrinsic space.

Fingerprinting as An Image Processing Problem

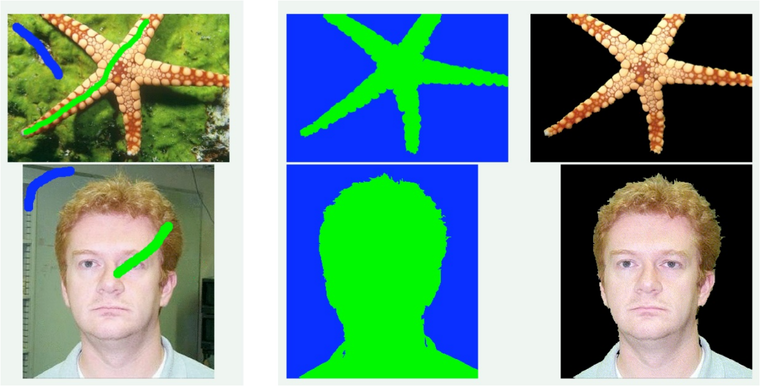


Figure : Source:
http://carbon.videolectures.net/v001/5d/luiumhtezps2i4tvfbomsfhuu3nt2vzu.pdf

Manifold Regularization (Belkin et (2006))

- A widely used tool for semi-supervised learning
- The intrinsic smoothness is

$$S(f) = \underbrace{\frac{1}{n^2} \sum_{i,j=1}^{n} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2}_{f^T \mathbf{L} f}$$

where w_{ij} represents similarity between fingerprints $\mathbf{x}_i$ and $\mathbf{x}_j$.

- Optimal solution $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i^*, \mathbf{x})$ where $\mathbf{I} = diag(\underbrace{1, 1, ..., 1}_{n})$,

$\mathbf{J} = diag(\underbrace{1, 1, ..., 1}_{l}, \underbrace{0, 0, ..., 0}_{n-l})$, $\mathbf{Y} = [y_1, y_2, ..., y_l, \underbrace{0, 0, ..., 0}_{n-l}]^T$,

$\mathbf{K} = \left[K(\mathbf{x}_i, \mathbf{x}_j)\right]_{n \times n}$, and $[\alpha_1, \alpha_2, ..., \alpha_n]^T = \left(\mathbf{JK} + l\lambda\mathbf{I} + \frac{l\gamma}{n^2}\mathbf{LK}\right)^{-1} \mathbf{Y}$.

Total Variation Regularization (Rudin et al. (1992))

- Instead of using Laplacian, use Total Variation (TV) for the intrinsic smoothness

$$S(f) = \sum_{i=1}^{n} \|\nabla f(i)\|_{L^p(w)}$$

where

$$\|\nabla f(i)\|_{L^p(w)} = \left(\sum_{j=1}^{n} w_{ij} |f_j - f_i|^p \right)^{1/p} .$$

- More effective than manifold regularization for image restoration/denoising in the area of image processing

Our interest

"Total Variation or Manifold Regularization better for semi-training of location fingerprints?"

TV Algorithm (Elmoataz et al. (2008))

1. Initial step: for $i, j \in [1, n]$

$$f_i^{(0)} = \begin{cases} y_i & \text{if } i \leq l \\ 1/2 & \text{otherwise} \end{cases} \quad (2)$$

$$\gamma_{ij}^{(0)} = w_{ij} \quad (3)$$

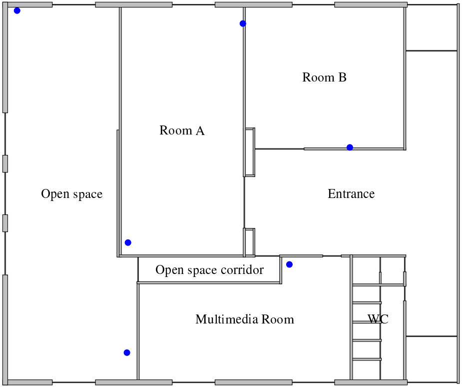
2. Iterative step: for $i, j \in [1, n]$

$$f_i^{(t+1)} = \begin{cases} y_i & \text{if } i \leq l \\ \frac{\sum_{j=1}^{n} \gamma_{ij}^{(t)} f_j^{(t)}}{\sum_{j=1}^{n} \gamma_{ij}^{(t)}} & \text{otherwise} \end{cases} \quad (4)$$

$$\gamma_{ij}^{(t+1)} = \frac{w_{ij}}{\|\nabla f^{(t)}(i)\|_{L^2(w)}} + \frac{w_{ij}}{\|\nabla f^{(t)}(j)\|_{L^2(w)}} \quad (5)$$

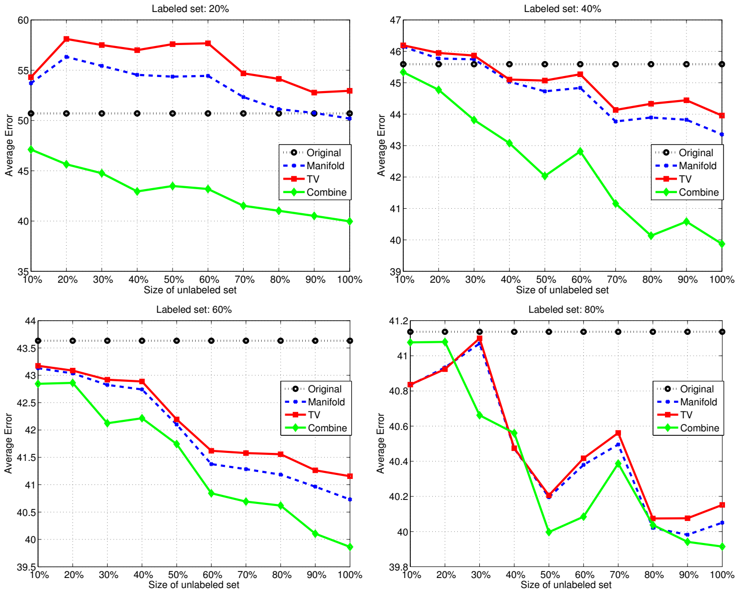
3. Stop when $|f^{(t+1)} - f^{(t)}| < \tau$ (predetermined threshold). The value of $f_i^{(t)}$ ($i > l$) is location estimate for fingerprint x_i.

Numerical Results: Wi-Fi Trento Dataset

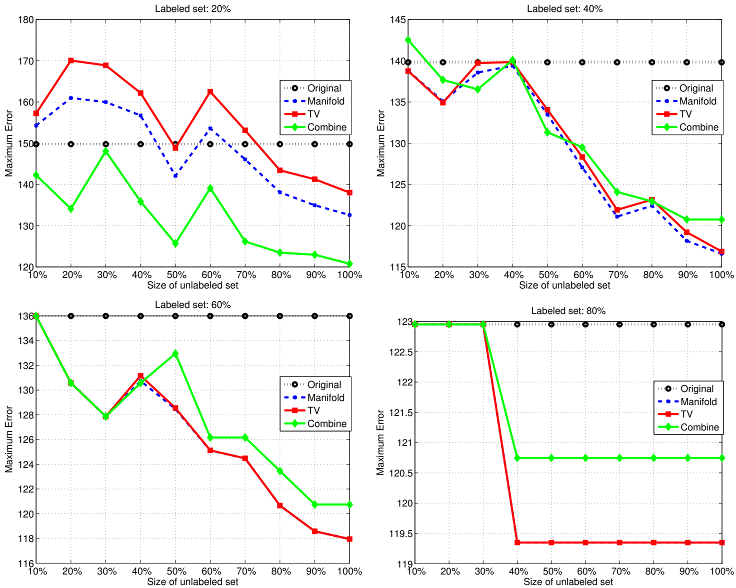


- 257 Wi-Fi RSSI fingerprints at 257 sample locations, 6 Wi-Fi APs
- Training set: 128 fingerprints (labeled and unlabeled)
- Testing set: 129 fingerprints
- Only use the intrinsic smoothness ($S(f)$) in the risk

Numerical Results: Average Location Error



Numerical Results: Max Location Error



Manifold vs. Total Variation Regularization

Our Finding

MR is better than TV for enriching training set of location fingerprints

[1]

[1] D. A. Tran and P. Truong, "Total variation regularization for training of indoor location fingerprints," in 2013 ACM MOBICOM Workshop on Mission- Oriented Wireless Sensor Networking (ACM MiSeNet 2013), Miami, Sep 2013.

Zero Training Set

- What if no training set to start with?
  - Totally avoid expensive survey phase
- Fingerprints become available in a stream manner, most of the time unlabeled: $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_t$, ...
- Minimal computing resource requirement for the positioning algorithm
  - Think “scalable", “energy", “lightweight"

Challenge

Need truly calibration-free online localization

Batching Approach

- Treat all the fingerprints obtained in the time window $[1, t]$ as the training samples
- Apply Manifold Regularization to find the best location predictor f

$$\min_f J(f) = \frac{1}{l} \sum_{i=1}^{t} \theta(y_i) L(f(\mathbf{x}_i), y_i) + \lambda \|f\|_K^2 + \gamma S(f), \qquad (6)$$

  - θ is the Heaviside step function.
  - $l = \sum_{i=1}^{t} \theta(y_i)$: number of labeled fingerprints observed

Disadvantage

- Infinite memory to store all the fingerprints
- Extremely slow to solve the minimization problem

(Partially) Incremental Approach

Pan and Yang (2007)

Two phases

1. Initial training: build a fingerprint map using the batching approach
2. Incremental update: Upon each new fingerprint, first update the weighted graph $W = \{w_{ij}\}$ and then run iterations of weighted averaging until convergence:

$$\forall\ i = 1, 2, ..., t:\ \ y_i^{new} = \frac{\sum w_{ij} y_j^{old}}{\sum w_{ij}} \tag{7}$$

Location of fingerprint $\mathbf{x}_t$ at the current time t will be $f(\mathbf{x}_t) = y_t^{new}$.

Disadvantage

- Faster than batching
- Still, infinite memory to store all the fingerprints

Our Approach: Truly Calibration-Free Online

Inspired by the results from online convex programming and its subsequent development in online manifold regularization.

- Model the online localization problem as an online convex programming problem

Online Convex Programming (Zinkevich (2003))

Convex Programming

- Given convex feasible set $\mathcal{F} \subset \mathbb{R}^n$, convex function $c : \mathcal{F} \to \mathbb{R}$
- Goal: Find $f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, c(f)$.

Online Convex Programming

- An infinite stream of cost functions, $c^{(1)}$, $c^{(2)}$, ..., $c^{(t)} : \mathcal{F} \to \mathbb{R}$
- Goal: find $f^{(t)} \in \mathcal{F}$ for each time t, not knowing cost $c^{(t)}$ which is revealed only after this selection, such that regret is minimal

$$\min \left\{ R(t) = \sum_{i=1}^{t} c^{(i)}(f^{(i)}) - \min_{f \in \mathcal{F}} \sum_{i=1}^{t} c^{(i)}(f) \right\}. \tag{8}$$

One can derive an algorithm based on gradient descent such that under mild conditions, $\lim_{t \to \infty} \frac{R(t)}{t} = 0$.

Online Manifold Regularization (Goldberg et al. (2008))

Ultimate goal: minimize

$$J(f) = \frac{1}{l}\sum_{i=1}^{t} \theta(y_i)(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_K^2 + \frac{\gamma}{t}\sum_{i,j=1}^{i} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2.$$

Let $p = l/t$ (“labeling rate"). Express $J(f)$ as a sum of convex functions

$$J(f) = \frac{1}{t}\sum_{i=1}^{t} \underbrace{\left(\frac{\theta(y_i)}{p}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_K^2 + \gamma\sum_{j=1}^{i} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2\right)}_{J^{(i)}(f)}$$

Transform to online convex programming

- Function space $\mathcal{F} = \mathcal{H}_K$
- Sequence of functions $c^{(1)} = J^{(1)}$, $c^{(2)} = J^{(2)}$, ..., $c^{(t)} = J^{(t)}$
- Goal: at time step t find function $f^{(t)} \in \mathcal{F}$ with minimal regret.

Online Manifold Regularization with a Buffer

Store only a finite "buffer" of previous fingerprints, indexed by a set $B^{(i)} \subset \{1, 2, ..., i-1\}$ so that the computation in time step i requires knowing the newly received fingerprint $\mathbf{x}_i$ and fingerprints in $B^{(i)}$.

Same online convex programming problem but replace cost function

$$J^{(i)}(f) = \frac{\theta(y_i)}{p}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_K^2 + \gamma \sum_{j=1}^{i} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2.$$

by a buffer-constrained cost function

$$J_B^{(i)}(f) = \frac{\theta(y_i)}{p}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_K^2 + \gamma \sum_{j \in B^{(i)}} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2.$$

Online Localization Algorithm (Tran and Zhang (2013))

Location estimate: $f^{(i+1)}(\mathbf{x}_{i+1}) = \sum_{j \in B^{(i+1)}} \alpha_j^{(i+1)} K(\mathbf{x}_j, \mathbf{x}_{i+1})$ where buffer $B^{(i+1)}$ and coefficients $\alpha_j^{(i+1)}$ are determined as follows:

1. Let

$$\begin{aligned} \beta^{(i)}_{j \in B^{(i)}} &= (1 - 2\eta_i \lambda)\alpha_j^{(i)} + 2\eta_i \gamma w_{ij}(f^{(i)}(\mathbf{x}_i) - f^{(i)}(\mathbf{x}_j)) \\ \beta_i^{(i)} &= 2\gamma\eta_i \sum_{j \in B^{(i)}} w_{ij}(f^{(i)}(\mathbf{x}_j) - f^{(i)}(\mathbf{x}_i)) - \frac{2\eta_i \theta(y_i)}{p}(f^{(i)}(\mathbf{x}_i) - y_i) \end{aligned}$$

2. Solve (using Matching Pursuit algorithm, time complexity $\mathcal{O}(b^3)$)

$$\begin{aligned} \underset{\alpha^{(i+1)}, B^{(i+1)}}{\text{minimize}} \quad & \left\| \sum_{j \in B^{(i+1)}} \alpha_j^{(i+1)} K(\mathbf{x}_j, .) - \sum_{j \in B^{(i)} \cup \{i\}} \beta_j^{(i)} K(\mathbf{x}_j, .) \right\|^2 \\ \text{s.t.} \quad & B^{(i+1)} \subset B^{(i)} \cup \{i\} \text{ and } |B^{(i+1)}| = b \text{ (given buffer size)} \end{aligned}$$

Calibration-Free Online Localization: Applications

Efficient use of GPS

For a GPS-equipped smartphone, GPS can be set to switch on once in a while and our algorithm can be used to compute the smartphone location during the GPS-free gaps $\Rightarrow$ energy saving.

Location assistant

In an indoor building, we can place location labels at popular locations (e.g,. info desks) which the phone can read automatically when passing nearby. These labels can be used by our algorithm to infer location at any other place.

Underwater tracking

AUV tracking is often based on built-in inertial navigation that has to dead reckon with GPS each time the AUV surfaces. Our algorithm could help better localize the AUV between these GPS fixes.

Summary

Effectiveness of fingerprint depends on richness of training data

Contributions

- The richer the training data, the more computation during the positioning phase; need a fast positioning algorithm
  - Finding 1: Positioning based on spatially hierarchical classification is many times faster than the conventional classification.
- If the training data is not rich, utilize unlabeled fingerprints
  - Finding 2: Manifold regularization works better than total variation regularization
- If no training data, need to learn on the fly
  - Finding 3: A truly calibration-free online localization based on online manifold regularization with buffer.

Acknowledgements

Our work was supported in part by the NSF award CNS-1116430. Any opinions, findings and conclusions or recommendations expressed in this material are ours and do not necessarily reflect those of the NSF.