AN EASY-TO-DEPLOY RFID LOCATION SYSTEM

Oranat Sangratanachaikul[1], Shin'ichi Konomi[2], Kaoru Sezaki[1,2]

Abstract

To complement the GPS, we can embed RFID tags in physical spaces and use them as location reference points. However, it is often costly to deploy such location reference tags over a large area. To minimize the deployment cost, we have developed an easy-to-deploy localization system in which location tags tell pedestrians where they are walking, and, in turn, pedestrians implicitly teach the tags where they are embedded in. In this paper, we describe the system's algorithm design, implementation, and the results of our experiments for dense and sparse configurations. Our system enables lightweight RFID deployment without explicit data inputs, thereby facilitating end-user design of pervasive location infrastructure.

1. Introduction

Many important human activities take place in urban and indoor spaces. However, it is difficult to support activities in such spaces by using the GPS technology alone. To support wayfinding, civilian safety, and emergency response in buildings, underground passages, and urban canyons, we need a usable and useful indoor location system that complements the GPS technology. Researchers proposed various indoor localization technologies [1] [3] [9] including RFID location tags [4] [7] [8] that require very small hardware cost.

System deployment is conceived as a major challenge in various pervasive computing projects. Place Lab[2] and PlaceEngine[5] for example, exploit existing Wi-Fi access points to enable large-scale location infrastructure with little hardware deployment cost. However, we must also consider the database deployment cost as RFID and Wi-Fi positioning systems usually require someone to create and maintain a mapping database that converts observed data (e.g., IDs and signal strengths) into location information. The cost to create and maintain such a database can become a major obstacle that hampers a location system. RFID location tags are easy to install, move, and remove, therefore their data cost can be a very serious issue.

Existing Wi-Fi localization projects [2] [5] reuse the data sets that organizations (e.g., companies and universities) maintain, exploit the "war-driving" data, or rely on *explicit* user contributions (PlaceEngine[5] explores a method to automatically capture relevant data from location-based queries as well). Approaches that rely on explicit voluntary contributions could potentially solve the data problem under certain social and cultural conditions.

To remedy the data problem without overly relying on contingent socio-cultural factors, we have developed a complementary system that exploits *implicit* user contributions. By simply walking around, users of our system simultaneously (1) obtain their location information from RFID tags, and (2) implicitly improve RFID tags' location information.

---

[1] Institute of Industrial Science, Univ. of Tokyo, 4-6-1, Komaba, Meguro-Ku, Tokyo 153-8505, Japan
[2] Center for Spatial Information Science, Univ. of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

2. Algorithm Design

We aim to provide robust, modular mechanisms that can be integrated with various deployment and usage models. Our system primarily considers two types of location tags: (T1) ones that are deployed by authorized land survey professionals and (T2) ones that are freely deployed by anyone. Location tags have *onstage* and *offstage* states: the system uses *onstage tags* to compute location information, and trains *offstage tags* until they are ready to "go on stage."

We now consider a physical space in which *onstage T1/T2 tags* and *offstage T2 tags* coexist. Let L be the location estimate of an *offstage tag*. Our system collects location information from the pedestrians who are in proximity to the tag, and incrementally computes L as follows:

$$L_{i+1} = \frac{(i \cdot L_i) + S_{i+1}}{i+1}$$

It obtains new location estimate L_{i+1} from pedestrian location S_{i+1} and existing location estimate L_i $(0 \leq i)$. This computational process can be triggered periodically, using the best pedestrian location S_{i+1} in each interval. Also, when there are multiple pedestrians nearby, S_{i+1} is a weighted sum of their location information. Note that our system currently uses RSSI (Received Signal Strength Indicator) to select the best S_{i+1} within each interval, and to assign a weight to each pedestrian. To estimate distances from radio signal strengths, the system considers the inverse square law that characterizes the spreading out of electromagnetic energy in free space.

An *offstage tag* is turned into an *onstage tag* when its error estimation becomes smaller than a threshold value. We estimate the error by using maximum likelihood estimator (MLE) with the assumption that the location S_i of nearby pedestrians follows a two-dimensional Gaussian distribution. This allows us to approximate the covariance matrix of the MLE, whose eigenvalues λ_1 and λ_2 are used to create an ellipse of error. Suppose the ellipse has semi-major axis a and semi-minor axis b:

$$a = -2ln(1-c)\sqrt{\lambda_1}$$

$$b = -2ln(1-c)\sqrt{\lambda_2}$$

We chose *0.95* for the value of confidence c, meaning that the ellipse contains the tag's real location with *95%* confidence. The tag's error estimation then is the area of the ellipse, i.e., πab.

Using the NS-2 network simulator [isi.edu/nsnam/ns/], we have verified the effectiveness of our approach for a setting in which 10 nodes move around in a *500m x 500m* two-dimensional space, based on Random Waypoint Model.

3. Implementation

We have implemented an RFID location system that utilizes the proposed algorithm to facilitate the system deployment. Our system consists of active RFID tags (RF Code™ Spider V), PC card-size RFID readers (RF Code™ Spider V Mobile Reader 303MHz), and linux notebook PCs. The tags announce their IDs every second.

Figure 1 shows the system architecture. Location Tag Manager (LTM) resides on each pedestrian's mobile device. LTM can handle any IDs that conform to an open RFID standard called *ucode*[8] . To convert *ucode* IDs into location information, we use Location Tag Database (LTDB) that resides on a wireless database server to which mobile clients access via ad-hoc communication networks.

The system handles *onstage* and *offstage* tags differently. When a mobile client detects *onstage* tags, it estimates its own location using the tags' IDs and RSSIs. When a client detects *offstage* tags, it estimates the tags' positions and errors based on the node location and the tags' update histories. It then updates LTDB using the estimated values. When a client doesn't encounter any tags more than a TTL (Time-to-Live) period, its location information is considered stale and not used to train *offstage* tags.

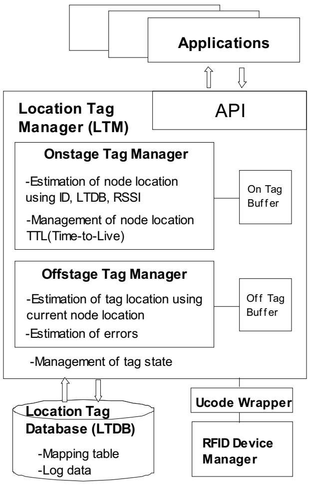
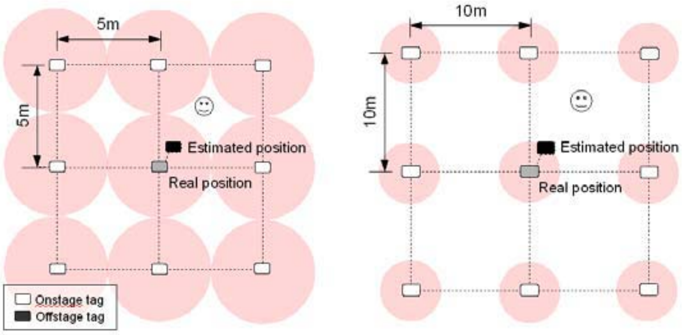


Figure 1: System architecture.

4. Experiments

The maximum communication range of our RFID tags is about *12 m* in a corridor of our building, and *2.5 m* on our outdoor athletic ground. Signal patterns vary as we place the tags near and away from the ground, walls and windows. We instrumented a small space (approx. *10m x 10m*) on the rooftop of our building with *5* RFID tags, and carried out a number of preparatory experiments to refine the parameters for the RSSI-based distance estimation, and adjust the time intervals for the incremental computation of location estimate *L*.

We then placed *9* tags on the athletic ground using the *5m* and *10m* grid configurations (see Figure 2), and measured the accuracy of our system's location estimation for *offstage* tags.



(a) *5m* grid configuration (b) *10m* grid configuration

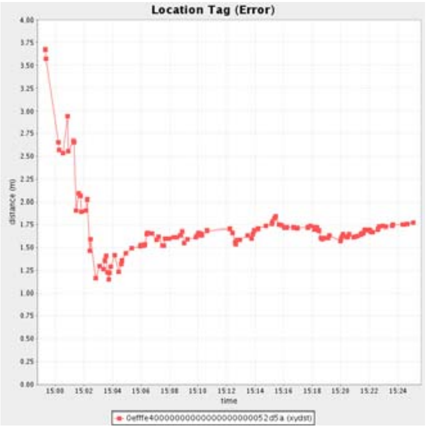
Figure 2: Placement of RFID location tags in the experiments. Red circles indicate the tags' approximate communication range (*2.5m*) on the athletic ground.

Figure 3 shows the result when a pedestrian walked for *26* minutes in the *5m* grid (a), and *38* minutes in the *10m* grid (b). In both cases, the difference between the real and the estimated positions of the *offstage* tag quickly decreases at an early stage and stays below *2m* thereafter. However, the error does not seem to converge to zero. We speculate that this could be partly because each tag has a different radio signal pattern depending on the way they are placed on the ground.

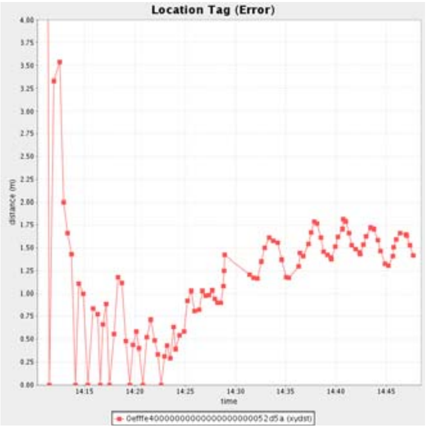
5. Discussion and Conclusion

Our system estimates the position of a newly installed tag without requiring explicit user inputs. The position estimation however is affected by the walking patterns of pedestrians as well as the variability and complexity of radio propagation. To understand how our system behaves in various environments, we are carrying out a number of additional computer simulations. We are also exploring the use of motion sensor-enabled RFID tags to detect modifications of tag placement and automatically re-estimate the new positions of the moved tags.

We plan to extend our localization framework for pervasive usage by considering mobile phones and other wireless gadgets with or without the RFID reader capability. In this context, we envision a scenario in which nearby pedestrian peers exchange their location information using wireless ad hoc communication. To improve the accuracy, we are exploring the uses of other pedestrian positioning technologies including the DRM (Dead Reckoning Module). Moreover, since scalability of Location Tag Database becomes a critical issue as the number of users increases, we are examining a possibility of a distributed architecture based on the Ucode Resolution mechanism[8] as well. The proposed approach could easily be extended to support various ID-based reference points including 2D barcodes, Wi-Fi access points, and Bluetooth/IR beacons.



(a) *5m* grid configuration



(b) *10m* grid configuration

Figure 3: The distance between the real and the estimated positions

6. References

[1] Bahl, P. and V. N. Padmanabhan (2000) Radar: An In-Building RF-Based User- Location and Tracking System, *Proc. IEEE INFOCOM*, pp. 775-784.

[2] LaMarca, A., Y.Chawathe, S.Consolvo, J.Hightower, I.Smith, J.Scott, T.Sohn, J.Howard, J.Hughes, F.Potter, J.Tabert, P.Powledge, G.Borriello, B.Schilit (2005) Place Lab: Device Positioning Using Radio Beacons in the Wild. *Proc. Pervasive 2005, pp. 116-133.*

[3] Ni, L.M., Y. Liu, Y.C. Lau, A.P. Patil (2003) LANDMARC: Indoor Location Sensing Using Active RFID, *Proc. First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pp. 407–415.

[4] Park, J.-M., J.-A. Kang, B.-G. Kim & Y.-S. Oh, Design of Ubiquitous Reference Points for a Location Information Service, *Proc. of the Second International Workshop on Ubiquitous Pervasive and Internet Mapping(UPIMap 2006)*, pp. 41-49.

[5] PlaceEngine Home Page. http://www.placeengine.com/en

[6] Sangratanachaikul, O. (2008) *Localization and Privacy Issues in RFID-Based Reference Point Systems.* Information Science and Technology M.S. Thesis, University of Tokyo, Japan.

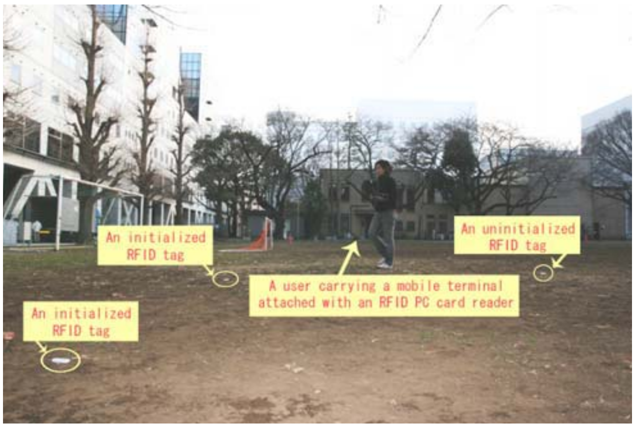
[7] Sezaki, K., S. Konomi (2006) RFID-based positioning systems for enhancing safety and sense of security in Japan, *Proc. of the Second International Workshop on Ubiquitous Pervasive and Internet Mapping(UPIMap 2006)*, pp.194-200.

[8] Ubiquitous ID Center Home Page. http://www.uidcenter.org

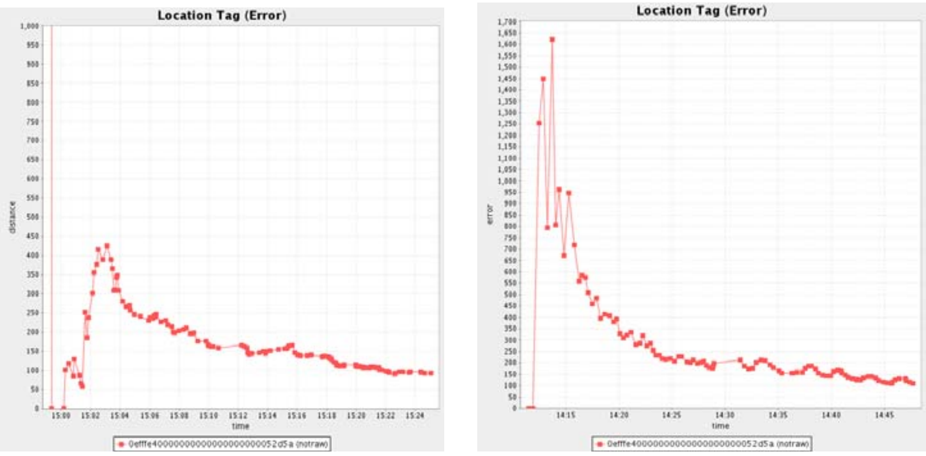
[9] Want, R., A. Hopper, V. Falcao, and J. Gibbons. (1992) The active badge location system. *ACM Transactions on Information Systems, 10*(1), pp. 91-102.

Appendix: Additional Information about the Poster

In our poster presentation, we will describe the system's algorithm design, implementation, and the results of our experiments using the four figures included in the paper as well as the following graphs and images.

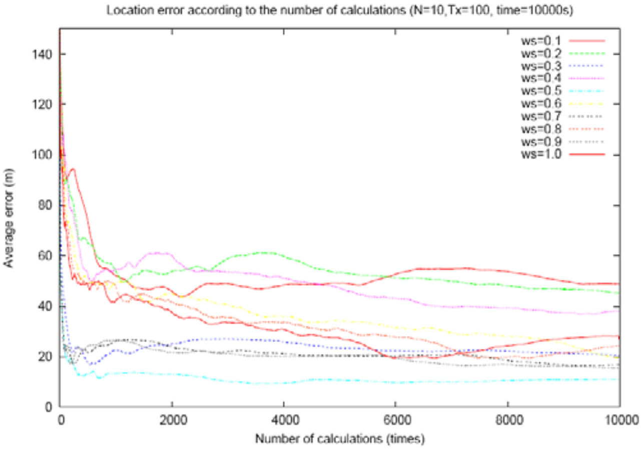


Experiment on athletic ground



(a) *5m* grid configuration (b) *10m* grid configuration

Error estimation using covariance matrices (without knowing the real position of the offstage tag); computed in real time during the experiments



Simulation setup:
Topology: 500m x 500m
Tag position: (250,250)
Tag communication range: 100m
Node initial position: random
Node communication range: 100m
Simulation time: 100m
Number of Nodes: 10
Mobility Model: Random Waypoint Model
(max speed: 50/s, pause time: 0.2s)

Simulation result showing average location error with varying window size (i.e., interval between the computation of L_i and L_{i+1}) from 0.1s to 1.0s