Positioning with Independent Ultrasonic Beacons *

Michael McCarthy and Henk Muller

Department of Computer Science, University of Bristol, U.K.
http://www.cs.bris.ac.uk/

Technical Report: CSTR-05-005

Abstract. In this paper we present a novel positioning technique that is based on transmissions from independent ultrasound beacons. These unsynchronised beacons emit narrow-band ultrasonic pulses and do not use any other medium, such as radio frequency or infra-red. A passive mobile positioning unit locates itself using the signatures of the received ultrasound signals. This unit measures shifts in the periodicities of the signals, a form of the Doppler effect, in order to estimate its location and velocity. We have initially tested the system using a simulator; the results suggest that the device is able to position itself with a 95% accuracy of 20 cm, and a 50% accuracy of 6 cm. Our previous experience is that these figures will degrade with the use of real hardware, but we aim for a 95% accuracy of better than 40 cm. The advantages of our system are three-fold: the infrastructure contains no wiring and, as such, can be easily retro-fitted with minimal aesthetic impact; it scales to any number of mobile positioning units; and the beacons are low-power, cheap and simple to construct.

1 Introduction

The problem of tracking mobile devices for use within context-aware applications is an important aspect of pervasive computing. Indeed, a number of different application domains – including gaming, tourism, health care, military and industry – are already benefiting from technologies that provide position information. While the Global Positioning System (GPS) has given application designers an opportunity to work with a system providing broad coverage, it is limited in its ability to perform in canyons and indoor environments. A number of researchers have presented systems that attempt to address this problem. These systems include radio-frequency based solutions such as ultrawide-band [1], 802.11 [2, 3] and GSM [4]; and those based on ultrasound such Active Bat [5] and Cricket [6]. Each of these solutions has strengths and weaknesses in terms of cost, set-up expense, accuracy and coverage.

We present the design of a positioning system that aims to be low-cost, has a minimal set-up and maintenance expense, and has a 95%-accuracy of 20cm. The system has been designed with single room coverage in mind but could be scaled to cover multiple rooms, as described in Section 5.

* Funding for this work is received from the U.K. Engineering and Physical Sciences Research Council as part of the Equator IRC, GR-N-15986.

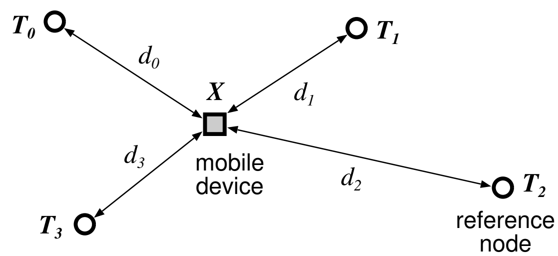


Fig. 1. Multi-lateration uses distances to reference nodes to find the location of a mobile receiver

The design is based on independent ultrasonic beacons. The motivating factor for independent beacons is that, in our experience of setting up and managing various systems, the wires linking devices within an infrastructure can be a nuisance. They are particularly undesirable in environments where aesthetics are important, such as a museum or a living room. In some instances, we have been able to hide the wiring within a suspended ceiling or some other ad-hoc method, but we believe the most effective solution is to create devices that are independent and that can be affixed unobtrusively.

An example of a system that has achieved this is the Cricket [7]. It employs independent beacons that transmit both radio and ultrasound signals. The radio signal is used to identify the beacon and to provide timing synchronisation, while the ultrasound signal is used to calculate distance.

Our system is meant to be simpler in design than the Cricket. Specifically, it is based on beacons that transmit ultrasound signals only. The advantage of not using RF is that we further reduce the cost, power consumption, and complexity of the beacon. The disadvantage is that we no longer have an obvious way of calculating distances, nor an obvious means of identifying signal sources. However, as we discuss in Section 2, we *are* able to identify signals, and, in Section 3, we show how we use them to calculate position. Section 4 describes our preliminary results taken from a realistic simulator. We hope to confirm these results once we have extended our experimental set-up to include a sufficient number of beacons.

2 Method

A large number of popular positioning systems in the literature employ techniques that use range data to calculate the position of a mobile device. This data is given as a set of measured distances from a mobile device to a number of reference nodes. Given that the locations of the reference nodes are known, it is possible to use a form of multi-lateration to calculate the position of the mobile device (Figure 1).

To measure these distances, most systems use the *time of flight* of a broadcast signal such as radio or ultrasound. In the case of the Global Positioning System, radio signals are transmitted by satellites orbiting the earth (these are the reference nodes). A mobile receiver decodes the time of transmission from within the signal and subtracts it from the signal's reception time. The resulting time-of-flight is multiplied by the speed of light to give the distance to the source of the signal.

Ultrasonic positioning systems work in a similar fashion to GPS; however, rather than use radio and the speed of light, they use ultrasound and the speed of sound through air. Some of these systems use a combination of radio and ultrasound [5, 6] to measure time-of-flight, while others use purely ultrasound [8].

The method that we employ is different. Specifically, we do not use a time-of-flight technique based on range information. Instead, we use shifts in the reception times of periodically transmitted signals as input to a form of *Doppler positioning*. The measured shifts are related to the movement of the mobile receiver, relative to the ultrasound beacons fixed within the infrastructure. (It is important to note that we use the Doppler-shift of the periodic signal, not the Doppler-shift of the ultrasonic carrier signal.)

2.1 Ultrasonic Beacons

Our system comprises a number of fixed ultrasonic beacons and one or more mobile receivers. The beacons are placed on the walls and ceiling of a room where positioning is to be performed. The beacons we have constructed use less than $100\mu A$ at $1.5V$. Each beacon is programmed to transmit a 10 cycle, 40 KHz ultrasonic pulse, a *chirp*, at regular intervals. The interval or period at which the pulses are sent is unique to the beacon; for example, beacon i transmits with a period of P_i. Unique periods make it possible for the receiver to distinguish chirp sources based on previous reception times. Typical transmission periods are on the order of 500 ms with differences between periods at around 5 ms.

We ensure that the base periodicities differ to such a degree that it is possible to uniquely identify beacons while the receiver is moving. The maximum velocity of the receiver is limited by the minimum distance between any two periods, P_i and P_j:

$$\frac{v\text{max}}{v_s} < \frac{1}{2}\frac{|P_i - P_j|}{\max(P_i, P_j)}$$

For example, if we choose P values of 500 and 505 ms, then, in order to identify beacons, the speed of the receiver must be less than 0.5% of the speed of sound, or $1.7ms^{-1}$. Limiting the speed to this value is only necessary in the absence of any prior knowledge about the receiver dynamics – with prior knowledge, the maximum speed is not an issue.

Upon reception of a chirp, the mobile device will measure a periodicity of P_i' for each beacon. This measured value will differ from the source beacon's periodicity P_i for two reasons. First, the beacon and receiver each have their own unsynchronised clock. This introduces a fixed offset in the measurement which is in the order of 10^{-6}. Second and more importantly, when the receiver is moving it will measure a periodicity $P_i + \Delta P_i$, where ΔP_i is a term introduced by the change in distance relative to beacon i. For example, if over the period P the receiver has moved 1 metre closer to the beacon, then the second pulse will arrive 3 ms earlier. It is this property that we exploit to do our positioning calculations.

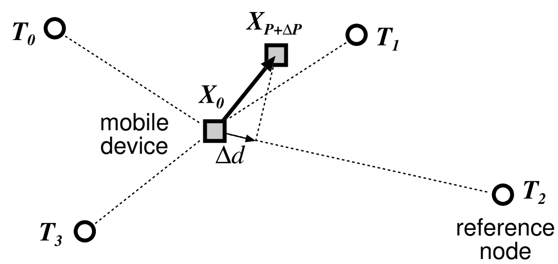


Fig. 2. To calculate position, our method uses the change in the relative distance to a beacon, over that beacon's period

2.2 Doppler Positioning

When the receiver is moving, shifts in the transmission periods are measured. If the receiver has moved towards a beacon, the received pulses will be compressed; if the receiver moves away, the pulses will become more separated. The amount that the pulses shift is proportional to the distance the receiver has moved over the period:

$$\Delta d = v_s \Delta P_i$$

Here Δd is the movement of the receiver relative to the beacon and v_s is the speed of sound, $\sim 343ms^{-1}$.

We can also calculate this distance by using the position of the beacon and the positions of the receiver at times 0 and $P_i + \Delta P_i$ (visualised in Figure 2):

$$\Delta d = \frac{(\boldsymbol{X_0} - \boldsymbol{T_i})}{|\boldsymbol{X_0} - \boldsymbol{T_i}|} \cdot (\boldsymbol{X_0} - \boldsymbol{X}_{P_i + \Delta P_i})$$

Hence,

$$v_s \Delta P_i = \frac{(\boldsymbol{X_0} - \boldsymbol{T_i})}{|\boldsymbol{X_0} - \boldsymbol{T_i}|} \cdot (\boldsymbol{X_0} - \boldsymbol{X}_{P_i + \Delta P_i}) \tag{1}$$

This formulation shows a relationship between our current location, our previous location, and two chirp reception times. Using Equation 1 and a sufficient number of readings, it is possible to iteratively estimate the receiver position.

We also note that it is more convenient to model the dynamics of the receiver in terms of position *and* velocity. To introduce velocity, we divide both sides of Equation 1 by $P_i + \Delta P_i$ and represent velocity as:

$$\boldsymbol{V} = \frac{(\boldsymbol{X_0} - \boldsymbol{X}_{P_i + \Delta P_i})}{(P_i + \Delta P_i)}$$

This is equivalent to the *average* velocity of our receiver over the time period $P_i + \Delta P_i$. With this, Equation 1 becomes:

$$v_s \frac{\Delta P_i}{P_i + \Delta P_i} = \frac{(\boldsymbol{X_0} - \boldsymbol{T_i})}{|\boldsymbol{X_0} - \boldsymbol{T_i}|} \cdot \boldsymbol{V} \tag{2}$$

Equation 2 relates the speed and location of our receiver to the measured shift in periodicity. In Section 3, we show how we implement this relationship using a Multi-hypothesis Kalman filter.

Our method is similar to Doppler positing systems that use shifts in RF signal frequencies to measure instantaneous relative velocities. Such a system is the Search and Rescue Satellite System (SARSAT) [9], which tracks distress beacons on board naval vessels. While our measurements for velocity are not perfectly instantaneous, they are sufficient as long as the transmission periods are not too large.

2.3 Collisions

The method described above assumes that we know the source of all received chirps. Because of the unique transmission periods, this assumption holds for most situations. If we consider the case where we know the source of all previous chirps and we know the position and velocity of the receiver, then we can predict when the next chirp for each beacon will arrive. However, a problem arises when two or more chirps are predicted to arrive at nearly the same time. This is indeed possible given that the beacons transmit at different periods and operate on separate clocks that drift with respect to one another.

The periods of the beacons have been chosen so that collisions will be sufficiently infrequent. If we define a collision as two chirps arriving within a 1 ms window and we use 10 beacons, the ether will be utilised for approximately $10/500$, or 2% of the time. As a first order approximation, 2% of that time will see a collision between two chirps. Collisions between more than two chirps are infrequent enough and can be ignored. Using ten beacons that transmit with 500 ms periodicity will give the system approximately one chirp every 50 ms, allowing for an update rate of 20Hz. Collisions will take out 1 in 50 chirps, or one measurement every 2.5 seconds.

It is important that the beacons have periodicities that are sufficiently different, and that they have a lowest common multiple which is quite high. If we assumed that two beacons have periodicities of 500 and 1000 ms, then these would either never collide (if they are out of sync), or they would always collide (if they are in sync). However, given that each beacon has its own clock, all periods P_i will be subtly different due to the cut of the resonator. Still, beacons with similar periodicities should be avoided as they will collide for a prolonged period of time once they are synchronised.

3 Algorithm

The tracking algorithm operates in two phases. The first phase is a short initialisation phase that is performed at start-up while the mobile receiver is stationary. The second phase is the positioning phase where a Multi-hypothesis Kalman filter tracks the receiver's position by continuously forking and trimming hypotheses based on the status of incoming chirps.

3.1 Initialisation

Before the algorithm begins tracking, it must first determine the transmission period, P_i, of each beacon and lock onto the incoming chirp train. These tasks are performed

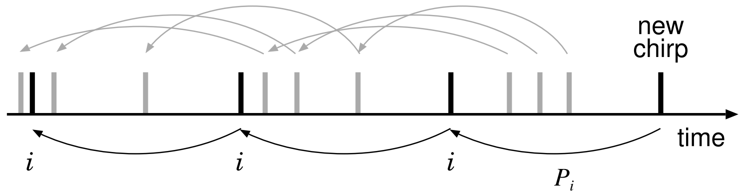


Fig. 3. The initialisation sequence scans the chirp history to find a source match for the most recent chirp

during the initialisation phase when the receiver is stationary. The process employs a type of auto-correlation where the chirp train is scanned in real-time for chirps arriving with similar periods. This is possible since the beacons each have their own unique transmission period. The algorithm scans the train history to find a suitable fit for the most recent chirp. If a fit is found, the chirp is labelled as belonging to the appropriate beacon, as illustrated in Figure 3.

The initialisation sequence takes around two minutes to complete for a receiver that has no record of the beacons, other than the expected chirp periodicity. The large time frame allows the algorithm to accurately estimate, using a large number of chirps, the transmission periods relative to the local clock. Good estimates of P_i are then saved to reduce the length of this phase for future positioning within the same room. Initialisation with saved period data only needs to lock onto the last few chirps, which takes less than 15 seconds.

3.2 Multi-hypothesis Kalman Filter

If it were possible to precisely determine the source of each of the chirps, we would be able to solve the problem using an Extended Kalman filter. However, ambiguity presented by collisions, reflections and noise requires us to implement another type of estimator that is better suited to these indeterminate situations. We have chosen to use a *Multi-hypothesis Kalman filter* [10, 11].

The filter uses a position-velocity model to track six different state variables: the three dimensional vectors $\boldsymbol{X}$ and $\boldsymbol{V}$. The measurement equation for the filter, Equation 2 derived in Section 2, relates position and velocity to the measured period shifts, ΔP_i.

$$\frac{(\boldsymbol{X} - \boldsymbol{T_i})}{|\boldsymbol{X} - \boldsymbol{T_i}|} \cdot \boldsymbol{V} = \frac{\Delta P_i \cdot v_s}{P_i + \Delta P_i}$$

Chirps are added to the filter as they are received by using a form of measurement integration called *single-constraint-at-a-time* (SCAAT) [12]. The SCAAT method allows us to keep our Kalman filters light-weight, so that many of them may run in parallel.

To provide a single estimate of position, the state of each of the hypotheses is given an equal weight and an average is taken. While it is possible to incorporate system covariances as weighting factors, we have found that since there are typically low numbers of hypotheses, taking the mean is sufficient. A discussion of the number of hypotheses is given in Section 4.

3.3 Forking and Trimming

When a chirp is received, the filter must first determine its source before it can be integrated. The source can either be a reflection or noise — in which case the chirp should be ignored — or one of the beacons in the room. As the filter is constantly tracking the position and velocity of the receiver, it is able to predict when chirps will arrive. We use the difference between the prediction and the arrival time of a chirp, scaled by the system covariance, to determine the likelihood of a particular source. In the literature, this value is referred to as the Mahalanobis distance or the χ^2 (chi-squared) statistic of measurement residuals [13].

If the χ^2 statistic for a beacon falls under a threshold then that beacon is considered a likely candidate for originating the chirp. Upon receipt of a chirp, there may be a number of beacons that fall under the threshold, we call this number M. M can be zero (for example if there is noise or an echo), one (in the case that the beacon is identified uniquely), or more than one (in the case that chirps are colliding) where we cannot be sure which beacon they belong to. In the latter two scenarios, the filter forks off $M+1$ hypotheses: one for each possible beacon, and an extra one that considers the chirp to be noise, and ignores it.

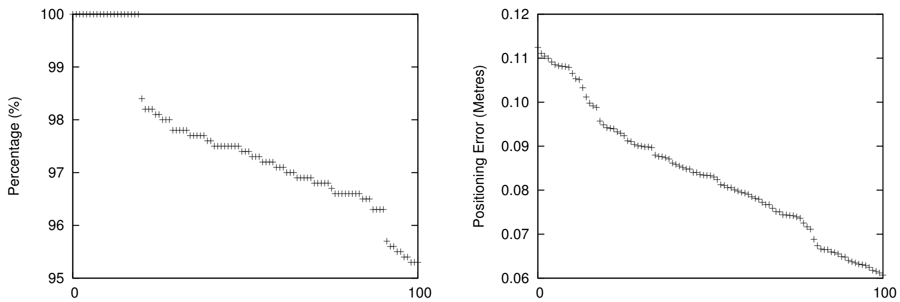
Hypotheses are forked based on the assumption that, as more chirps arrive, it will become evident which hypotheses are based on correct decisions. The incorrect hypotheses must then be trimmed from the list to prevent an overuse of resources. Trimming takes two forms: hypotheses that converge on a similar solution are combined into a single solution, and hypotheses that are "bad" are removed from the system. This last step is a potentially dangerous step, as we do not want to accidentally remove the correct solution, and it is not always obvious what the correct solution is. We use three heuristics to identify bad solutions:

- The mean χ^2 statistic
- The confidence volume derived from the system covariance
- The number of beacons that a solution has ignored

The first two heuristics are intuitively good; however, it is very easy for a hypothesis to lock onto a wrong solution, if it simply ignores some of the beacons. In particular, if a solution starts to use only five beacons, there is a myriad of solutions with low χ^2, as the system is under-constrained with fewer than six beacons. Hence, we find that the number of beacons used in creating the position estimate is a good heuristic.

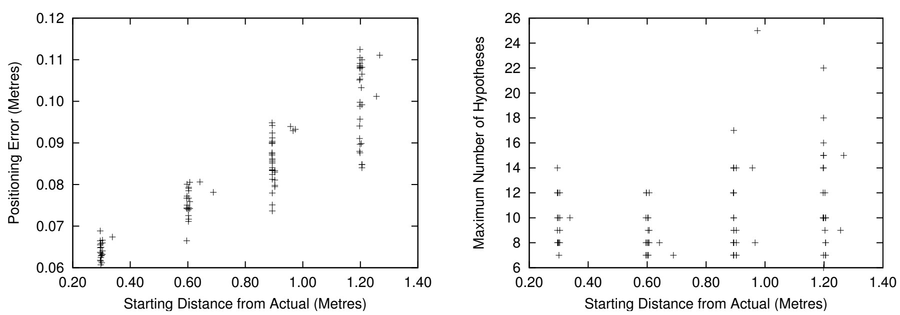
4 Results

In order to evaluate the model described above, we have employed a simulator used in the development of a previous positioning system [8]. The simulator controls a number of different environmental variables, including noise, occlusions, reflections, the number and location of beacons, and the receiver path. It is an invaluable tool in that



(a) (b)

Fig. 4. Position quality results for 100 trials (a) sorted on percentage of readings below 0.5 metre error, (b) sorted on mean distance from actual position



(a) (b)

Fig. 5. The effects of starting position on (a) position quality and (b) number of hypotheses

it presents us with an evaluation method where the ground truth is precisely known. This is important since the measurements that we work with contain no data to identify chirp sources, making debugging of our system using only real-world measurements near impossible. Also, the simulator allows us to easily assess the performance of our algorithms under varying environmental characteristics that are difficult to control physically. These include noise and occlusions as well as reflections. While our current hardware does not have enough beacons to include measurements in this paper, we are at present extending it to include more beacons.

The control variables that we varied during our simulations are:

– number of beacons
– frequency of occlusions
– frequency and number of reflections
– receiver start position

We conducted a few thousand simulation trials to test the effects of the control variables on the positioning performance of our algorithm. The system needs at least seven

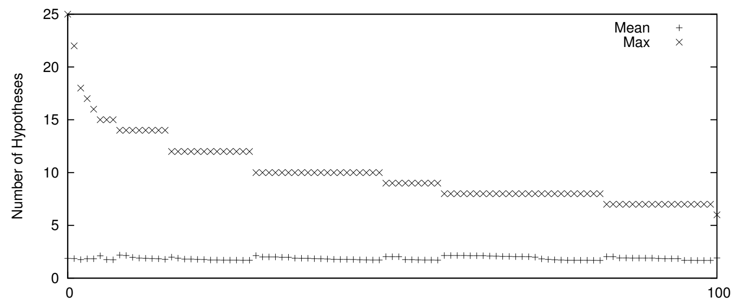


Fig. 6. Maximum and mean number of hypotheses for 100 ten minute trials (sorted on the maximum number of trials)

beacons in order to operate. Intuitively, we expected the minimum to be six, but in the presence of collisions, using six beacons means that the equations are underspecified sufficiently to cause the algorithm to diverge. Six beacons also make it very difficult to cope with echoes and occlusions.

For the results given in Figure 4 we use eight beacons to track position for 100, ten minute trials. The 100 trials are a selection of scenarios with occlusion frequency ranging from 0 to 10%, reflection frequency from 0 to 50%, and start position from 0 to 1.2 metres. As the figure shows, the output position stays within 0.5 m of the true position better than 95% of the time. The average distance from the true position is less than 12 cm.

Interestingly, we have not observed a relationship between occlusion frequency, reflection frequency and the positioning accuracy. We believe that this is evidence that the multi-hypothesis strategy is performing as desired. There is a relationship, however, between the receiver's assumed start position and the position quality, as shown in Figure 5(a). It appears that, as the assumed start position moves away from the actual start position, positioning errors increase. We are planning to run longer trials to assess whether, over time, the accuracy improves. Figure 5(b) also shows that the maximum number of hypotheses increases in the same fashion. We attribute these observations to the nature of our positioning equations. As these equations are iterative, they assume that previous estimates for position are correct. While the effect is not fatal, our results stress the importance of having a good starting position. One way to ensure this is to seed the algorithm with a number of different hypotheses, each having a different start position.

In terms of resource usage, Figure 6 shows the maximum and mean number of hypotheses used during the trials. While the mean number of hypotheses remains below three, there are instances where the number gets as high as 25. Large numbers of hypotheses result when the algorithm is unable to distinguish the correct hypothesis from the pool of hypotheses. Because the number of hypotheses (at least) doubles every time a chirp is received, the pool can, quite quickly, explode into an unmanageable size. For this reason, it is important that incorrect hypotheses be identified as early as possible, to prevent them from spawning more incorrect hypotheses.

We have executed our algorithm on a 200 MHz Gumstix wearable [14]. Executing one hypothesis takes around 7% of the available CPU time, hence we can, when there is a demand, execute up to 14 hypotheses simultaneously. Our present algorithm requires more than 14 hypotheses on 8% of the trials, but over the total run of all trials the number of hypotheses only exceeds this maximum value for 41 out of 914000 chirps which is 0.004% of the time. We are working on strategies to bring the number of hypotheses down so that it can work in real time in all cases. Reducing the number of hypotheses is also important to leave more time on the wearable to run the application program.

5 Future work

One consideration for future work is the extension of the system for use in multiple rooms or an entire building. A large building will require the installation of hundreds of beacons to provide sufficient coverage. We cannot give each beacon a unique periodicity in this case, since the minimum difference between periods would mean that the largest periodicities will be several seconds, which is too large to be of practical use.

Instead, we propose to use a set of, say, 20 different beacon periodicities only. Spacing beacons by 5 ms will mean that the most infrequent beacons will have a periodicity of 600 ms. If one picks beacons at random, and distributes them over rooms and corridors in a building, then from most locations one will observe a unique subset of the 20 different beacons. We can use a global finger printing technique to find out what subset we see, followed by precise tracking using the algorithms described earlier. This should overcome the problem of transitions between rooms where new beacons are discovered and old ones are lost.

For the solution detailed in this paper, we have used a Multi-hypothesis Kalman filter. The spawning and trimming of hypotheses is a delicate process, as we do not want to accidentally lose the correct solution. Instead of using a Kalman filtering approach, the multi-modal nature of the problem lends itself very well to the use of a *particle filter* [15]. Initial studies have shown that a particle filter is able to estimate position, and we are at present studying the stability of the estimates produced by the particle filter.

6 Conclusions

We have presented a method for estimating position using a network of unsynchronised beacons. The beacons each transmit ultrasonic pulses at unique periodic intervals. A mobile device measures the deviation in periodicities (due to a Doppler shift), and estimates its position using a Multi-hypothesis Kalman filter. Because the unsynchronised beacons occasionally transmit signals simultaneously, multiple hypotheses are required to disambiguate the input signals. Using our simulator we have estimated a 95%-accuracy of 20cm.

The accuracy is lower than that of other ultrasonic positioning systems that use some synchronisation mechanism. We believe, however, that this is a price worth paying for a system that can be retrofitted to existing buildings without a large aesthetic impact. This is especially important if we want to fit our systems in museums or living rooms, for example, where we have previously received objections from the owners about unsightly wires. The absence of RF makes our system very cheap to build, and low power.

References

1. Ubisense Limited: Website. http://www.ubisense.net/ (2005)
2. Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: INFOCOM (2). (2000) 775–784
3. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I.E., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.N.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In Gellersen, H.W., Want, R., Schmidt, A., eds.: Pervasive. Volume 3468 of Lecture Notes in Computer Science., Springer (2005) 116–133
4. Otsason, V., Varshavsky, A., LaMarca, A., de Lara, E.: Accurate GSM Indoor Localization. In Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H., eds.: Ubicomp. Volume 3660 of Lecture Notes in Computer Science., Springer (2005) 141–158
5. Ward, A., Jones, A., Hopper, A.: A New Location Technique for the Active Office. In: IEEE Personnel Communications, volume 4 no.5. (1997) 42–47
6. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: Mobile Computing and Networking. (2000) 32–43
7. Smith, A., Balakrishnan, H., Goraczko, M., Priyantha, N.: Tracking Moving Devices with the Cricket Location System. In: Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM Press (2004) 190–202
8. McCarthy, M., Muller, H.L.: RF Free Ultrasonic Positioning. In: Seventh International Symposium on Wearable Computers, IEEE Computer Society (2003)
9. International Satellite System for Search and Rescue: Website. http://www.cospas-sarsat.org/ (2005)
10. Kalman, R.E.: A New Approach to Linear Filtering and Prediction. In: Journal of Basic Engineering (ASME). (1960) 82(D):35–45
11. Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Academic Press Professional, Inc., San Diego, CA, USA (1988)
12. Welch, G., Bishop, G.: SCAAT: Incremental Tracking with Incomplete Information. In: SIGGRAPH 97 Conference Proceedings, Annual Conference Series. (1997)
13. Grewal, M.S., Weill, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, and Integration. Wiley, Inc. (2001)
14. Gumstix Inc.: Website. http://www.gumstix.com/ (2005)
15. Hightower, J., Borriello, G.: Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In Davies, N., Mynatt, E., Siio, I., eds.: Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp 2004). Volume 3205 of Lecture Notes in Computer Science., Springer-Verlag (2004) 88–106