A New Method for Auto-Calibrated Object Tracking *

Paul Duff[1], Michael McCarthy[1], Angus Clark[1], Henk Muller[1], Cliff Randell[1], Shahram Izadi[2], Andy Boucher[3], Andy Law[3], Sarah Pennington[3], and Richard Swinford[3]

[1] Department of Computer Science, University of Bristol, U.K.
[2] Microsoft Research, Cambridge, U.K.
[3] Royal College of Art, U.K.

duff@cs.bris.ac.uk

Abstract. Ubiquitous computing technologies which are cheap and easy to use are more likely to be adopted by users beyond the ubiquitous computing community. We present an ultrasonic-only tracking system that is cheap to build, self-calibrating and self-orientating, and has a convenient form factor. The system tracks low-power tags in three dimensions. The tags are smaller than AAA batteries and last up to several years on their power source. The system can be configured to track either multiple near-stationary objects or a single fast moving object. Full test results are provided and use of the system within a home application is discussed.

1 Introduction

Position sensing is an important aspect of pervasive computing. As a form of context input, it provides context-aware applications with the ability to model relationships between users and their environment. With this information, applications can provide a number of useful interactions in scenarios ranging from an audio tour-guide to an augmented reality application.

Narrowband ultrasonic positioning is an attractive form of positioning because it is low cost. Transducers are cheap and readily available, and expensive, high-precision oscillators are unnecessary because ultrasonic signals travel relatively slowly when compared to other signals such as RF. There are a number of narrowband ultrasound systems described in the literature that compute positions using a variety of different techniques. Some, such as the Active Bat [1], perform position *tracking* where a single governing application tracks multiple objects within a framework. Others perform *positioning*, where each object calculates its own private position, such as the systems developed at the University of Bristol [2, 3].

* 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.

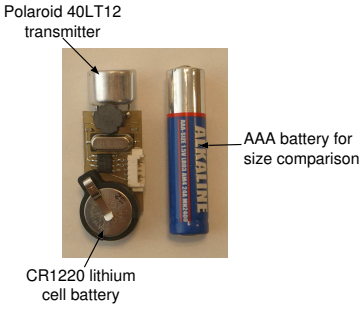


Fig. 1. Ultrasonic Tag

The issue of cost addressed by these applications, however, does not consider the expense of setting up and configuring the infrastructure supporting them. The uptake of positioning technologies by users outside of the ubiquitous computing community relies on us making them *accessible*. This term refers to all costs surrounding the technology:

- component cost;
- configuration expense;
- power consumption;
- form factor.

We have developed an object tracker that addresses each of these issues. Unlike most systems that utilise RF and ultrasound, including the Active Bat and Cricket [4], our system uses only ultrasonic signals. The absence of RF circuitry improves the power consumption, component cost, and form factor of tags tracked by the application. Our prototype tags are smaller than AAA batteries, run on 3V CR1220 cell batteries (see Fig. 1), and weigh 8 grammes (including CR1220 battery). We have also developed auto-calibration and self-orientation applications that greatly reduce the time and expertise needed to set-up the infrastructure, making it easier for artists and non-experts to use. Furthermore, the system can be configured to track either multiple near-stationary objects or a single fast moving object. The total component cost for the system, including the infrastructure and one tag, is around $100.

We explore the use of the ultrasonic system for tracking objects as they move around the home. In this test, non-technical researchers installed and used the system without any 'expert' help. A discussion of the use of the tracker as a part of this project is presented at the end of the paper.

2 System Description

The tracking system consists of a set of fixed receivers and one or more transmitting tags. The tags are each powered by a 3V coin cell battery. A PIC microcontroller on each tag controls the periodic activation of a 40kHz narrowband

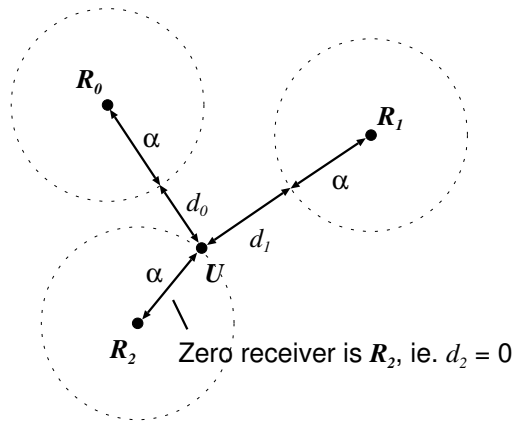


Fig. 2. Ultrasonic tracking. $\boldsymbol{U}$ is the position of the transmitter tag, $\boldsymbol{R}_0$, $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ are three receiver positions

ultrasonic transducer. In the case of a single object tracking system, the tag chirps regularly with an operational life of several months. In the case of multiple objects that are to be tracked individually, each tag chirps with a specific signature (described in detail in Sect. 5.1). The operational life increases to several years because chirping is less frequent.

The receivers are connected to a controller which determines the relative timing of the input signals. The controller sends the recorded timing information over a serial cable to a PC running the tracking software. The system requires at least four receivers in order to locate objects (discussed in Sect. 3). We use six for the system described here in order to provide some redundancy and increase accuracy. The maximum range of the tags is around 8m; this is sufficient to provide coverage for a room of 6×6m, for example.

When a signal emitted by a tag is detected by the nearest receiver, it is assigned a relative distance of zero metres. As the chirp travels outward from its source, the other receivers detect it after an additional delay. By factoring with the speed of sound, these delays can be converted into relative distances. Hence for a system using m receivers, the distances can be expressed as follows:

$$d_i + \alpha = |\boldsymbol{R}_i - \boldsymbol{U}| \qquad (0 \leq i < m) \tag{1}$$

Here d_i is a relative distance measured using receiver position $\boldsymbol{R}_i$, $\boldsymbol{U}$ is the origin of the chirp, and α is an unknown offset. Note that the nearest receiver $\boldsymbol{R}_z$ will always give a relative distance $d_z = 0$. We refer to receiver $\boldsymbol{R}_z$ as the *zero receiver*. Figure 2 shows a 2D example of this arrangement.

The constant offset α can be eliminated by observing that for the zero receiver $\boldsymbol{R}_z$:

$$\alpha = |\boldsymbol{R}_z - \boldsymbol{U}| \tag{2}$$

Hence we are left with a system of equations in terms of fixed receiver positions $\boldsymbol{R}_i$, the chirp origin $\boldsymbol{U}$, and measured relative distances d_i. We assume a fixed

speed of sound, though it is possible to estimate this at the cost of an additional receiver [5]. Note that this is only feasible once the system is calibrated.

$$d_i = |\boldsymbol{R}_i - \boldsymbol{U}| - |\boldsymbol{R}_z - \boldsymbol{U}| \qquad (0 \leq i < m, i \neq z) \tag{3}$$

In the next section we describe how we determine the positions of the tracking system's fixed receivers $\boldsymbol{R}_i$ by auto-calibration, using an approach similar to calibrating a positioning system [9]. In Sect. 4 we describe the auto-orientation algorithm, while in Sect. 5 we examine two alternative methods for determining the chirp origin $\boldsymbol{U}$.

3 Auto-calibration

Tracking is only possible once the positions of the receivers $\boldsymbol{R}_i$ are known. As we do not want to rely on people surveying the space by hand, our aim is to determine the positions of $\boldsymbol{R}_i$ algorithmically. To accomplish this, we take a single transmitting tag and move it around the room, taking care to collect distance readings from all parts of the room. This stage takes around 30 seconds, in contrast to taking measurements manually which can take 20 minutes or more, especially when receivers are mounted out of reach. In our experience, manual measurements require the work of two people, and often involve reaching awkwardly and noting down large numbers of distances and offsets. Auto-calibration obviates the need for this inconvenient process.

Once we have a representative set of distance measurements, we aim to solve (4), where m receivers and c tag positions are used, and only distances d_{ij} are known. There are thus m receiver positions $\boldsymbol{R}_i$, c zero receivers $\boldsymbol{R}_{z_j}$, and c tag positions $\boldsymbol{U}_j$ to be determined.

$$d_{ij} = |\boldsymbol{R}_i - \boldsymbol{U}_j| - |\boldsymbol{R}_{z_j} - \boldsymbol{U}_j| \qquad (0 \leq i < m, i \neq z_j, 0 \leq j < c) \tag{4}$$

A solution can only be found if there is more information known about the system than is unknown and to be computed—the system must be *over-specified*. Each unknown receiver position has three degrees of freedom in 3D space that must be found. Additionally, each new tag position contributes a further three unknowns. This is balanced by $m - 1$ relative positions and one zero receiver index gained with each new chirp transmission.

Six degrees of freedom can also be eliminated by describing the planar relationship between the first receiver and two others. By fixing the co-ordinates of one receiver, we set an origin for the 3D space in which solutions will be computed. The remaining three co-ordinates set the axes for the 3D co-ordinate system. In this way we avoid a situation where the same solution could be rotated or translated to produce an infinite number of solutions. Setting the axes leads to an arbitrarily rotated solution, a problem that is solved in Sect. 4.

The approximate constraint generalised to a system of m receivers and c chirp transmission positions is:

$$3(c + m) - 6 < (c - 1)m \tag{5}$$

The constraint is an under-bound; in reality an excess of known data is needed for good results. Readings from the same location are redundant as they will not contribute any additional information to the system. The algorithm must also be able to handle noisy measurements from the ultrasonic sensors.

It is clear from (5) that we must use a minimum of four receivers in order to over-specify the system. In the case of our six receiver example, this constraint suggests that at least six sets of relative distance data are needed, though more are desirable.

Given perfect data, it would be possible to find perfect solutions for the receiver and tag positions, balancing (4). In practice this is not possible, since distance readings have a limited granularity and are subject to noise and distortions. We can only hope to minimise the overall error and find a *best fit* for the unknown values.

Taking a least squares approach to error minimisation, we express the problem in terms of a minimisation function F, as shown in (6).

$$F = \sum_{i=0,\ i \neq z_j}^{m-1} \sum_{j=0}^{c-1} \left(d_{ij} - |\boldsymbol{R}_i - \boldsymbol{U}_j| + |\boldsymbol{R}_{z_j} - \boldsymbol{U}_j| \right)^2 \tag{6}$$

We use the Levenberg-Marquardt nonlinear least squares fitting algorithm [6], as it is reasonably suited to problems of high dimensionality. Attempts using other techniques such as Simulated Annealing [7] revealed that they converge less reliably on good solutions. We use an implementation provided within the GNU Scientific Library (GSL) [8], and refer to this central part of the algorithm as the *solver*.

As a randomised search process, the solver is not guaranteed to produce a good solution on any one occasion. We therefore execute the solver multiple times, using different seed values and input data to increase the probability of finding receiver positions accurately. The most promising solutions can then be selected according to quality heuristics. This process is repeated in several rounds, where each round applies more demanding criteria for convergence than the previous one.

At the end of a round, we select the most promising 10% of the candidate solutions [9]. The best solutions are taken forward to the next round, and the process repeated until there is a final set of candidate solutions remaining. In this way we survey the search space broadly without a prohibitive computation cost.

The remaining candidate solutions are then each represented in terms of the distance between respective pairs of receivers. This representation is independent of the origin and rotation of the solution, and hence can be used to compare different solutions. We can expect that two geometrically similar solutions will share similar distances between corresponding receiver pairs, even if one solution is rotated with respect to the other. There are 15 distances between the six receivers, hence we describe the solution as a point in a 15-dimensional space. We found that good solutions cluster around one point in this 15-dimensional space, while poor solutions are scattered at lower density in other arrangements.

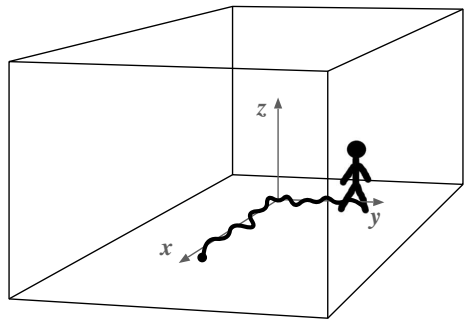


Fig. 3. The user walks along an L-shaped path, defining the x and y axis of the co-ordinate system.

By performing a greedy clustering algorithm on these distances we identify where the correct solution lies.

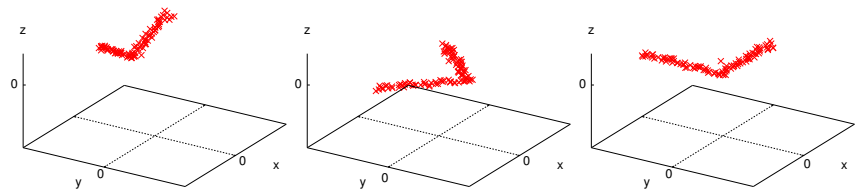
4 Orientating an Auto-calibrated Solution

Once the system has selected a best solution for the positions of the receivers, we find that the axes of the solution are placed arbitrarily. This is inherent to the problem, as we do not constrain the use of a particular origin or orientation. Specifically, the solver is allowed to invent any co-ordinate system it sees fit, provided it is orthogonal. As such, these axes may appear to bear little resemblance to the layout of the room. In order to place the receivers within a pre-defined co-ordinate system—for example one where the axes align conveniently with the walls and floor—we allow the user to specify the co-ordinate frame that they would like to use through an auto-orientation stage.

In keeping with the principle of ease of deployment, the method automates much of the process and requires no technical knowledge on the part of the user. All the user must do is take a single transmitting tag in hand and walk along an L-shaped path which defines the axes that they would like the system to use, as shown in Fig. 3.

The system collects tag positions as the user walks along the path. This set of position data defines the L-shape path in terms of the internal co-ordinate frame used by the system, as shown in Fig. 4(a). A two stage search process is then applied to the position data, in order to determine the transformation that aligns the L-shape path to the axes of the internal co-ordinate frame. Splitting the search into two stages greatly simplifies the process and improves the reliability of the system.

The first stage borrows techniques from Principal Components Analysis [10] and attempts to find the best fitting plane through the set of training points. Let $\boldsymbol{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, ..., \boldsymbol{X_n}\}$ be the set of tag positions. We first calculate the mean, $\bar{\boldsymbol{X}}$, and covariance matrix $\boldsymbol{\Sigma}$ of the data set $\boldsymbol{X}$. The eigenvectors, $\boldsymbol{e_1}$,



(a) (b) (c)

Fig. 4. The L-shaped path defined by user. (a) shown in the internal co-ordinate system, (b) transformed onto the XY-plane, (c) aligned with the axes.

$\boldsymbol{e_2}$, $\boldsymbol{e_3}$ of the covariance matrix define an alternative orthogonal basis. If we order the eigenvectors by their corresponding eigenvalues, each axis accounts for a diminishing proportion of the variance present in the data. In our system the first two eigenvectors will lie on the plane defined by the L-shaped path (ie. the XY-plane), while the third eigenvector accounts for any variations in height along the path (ie. the z-axis).

Converting the points to this new co-ordinate system, we get:

$$\boldsymbol{X'_i} = [\hat{\boldsymbol{e_1}}, \hat{\boldsymbol{e_2}}, \hat{\boldsymbol{e_3}}]^T (\boldsymbol{X_i} - \bar{\boldsymbol{X}}) \tag{7}$$

where $\hat{\boldsymbol{e}}$ denotes the normalised vector $\boldsymbol{e}$.

We have now arrived at a co-ordinate system where the L-shape lies on the XY-plane, and with height represented by the z-axis, as shown in Fig. 4(b). The next stage is to find the translation $\boldsymbol{T} = [t_x, t_y, 0]^T$, and the rotation around the z-axis, $\boldsymbol{R_z}(r_z)$, that will align the L-shape path with the x and y-axis, and with the corner located at the origin.

In order to achieve this, we first project the set of points $\boldsymbol{X'}$ onto the XY-plane, thereby removing any height variations present along the L-shaped path. Next we apply an adaptive simulated annealing algorithm [7, 11] that minimises the following energy function.

$$E(t_x, t_y, r_z) = \sum_{i=1}^{n} e_{xi}^2 \times e_{yi}^2 \tag{8}$$

where

$$\begin{bmatrix} e_{xi} \\ e_{yi} \\ 0 \end{bmatrix} = \boldsymbol{R_z}(\boldsymbol{X'_i} - \boldsymbol{T}) \tag{9}$$

The energy function captures the notion that the L-shape should be aligned to the axes. However, the system has no knowledge as to which section of the L-shape should be located on the x-axis, and which should be located on the y-axis. We therefore examine the start and end points of the L-shape to ensure that they lie on the positive x-axis and positive y-axis respectively, updating the rotation $\boldsymbol{R_z}$ as required ($+90^o$, $+180^o$ or $+270^o$).

The transformations derived above can be combined to give a single homogeneous transformation. The result yields a transformation that converts points from the internal co-ordinate frame used by the system to the co-ordinate frame defined by the user, as shown in Fig. 4(c). This transformation is applied to all subsequent points thereby providing position information in the co-ordinate frame specified by the user.

The current implementation assumes the L-path specified by the user lies on the XY-plane. The system could easily be extended to allow the user to specify which co-ordinate plane (XY, XZ or YZ) the L-shape should reside. However, this introduces further complexity for the user and in most cases will be unnecessary.

5 Tracking Objects

Our system can be used to track multiple objects with a low resolution in time, or a single object with a high time resolution (33Hz).

5.1 Multiple Objects

A fully calibrated system can be used to track one or more ultrasonic transmitters. It is often desirable to track multiple objects, each with a transmitting tag attached. In order to track individual objects reliably, each tag needs to transmit a signature that uniquely identifies the source, separating it from background noise and signals from other tags.

Estimator In order to determine the position $\boldsymbol{U}$ of a tag, we solve the system of equations described in (10).

$$d_i = |\boldsymbol{R}_i - \boldsymbol{U}| - |\boldsymbol{R}_z - \boldsymbol{U}| \qquad (0 \leq i < m, i \neq z) \tag{10}$$

The receiver positions $\boldsymbol{R}_i$ are known from the auto-calibration stage, and relative distances d_i are measured by the system when the tag transmits. The zero receiver $\boldsymbol{R}_z$ is determined trivially by checking the relative distances recorded by each receiver. To solve the system of equations for $\boldsymbol{U}$, we use a least-squares minimisation process very similar to that described in Sect. 3. However, since there is much more known information in the system, we only need to run the Levenberg-Marquardt algorithm once to find a solution.

As we have opted to use cheap ultrasonic transmitters with a narrow transmission frequency range, it is not feasible to encode the signature by altering the frequency of the transmitted ultrasound wave. Instead, the tags are configured to transmit three short pulses in quick succession. The time between transmission of each of the pulses is selected from a set of six discrete durations, varying from 30-40ms. These durations are permuted so that each tag has a unique timing signature, as illustrated in Fig. 5.

We expect that the computed positions of a tag will be virtually the same for each of the three pulses, as the tag cannot move very far between pulses.

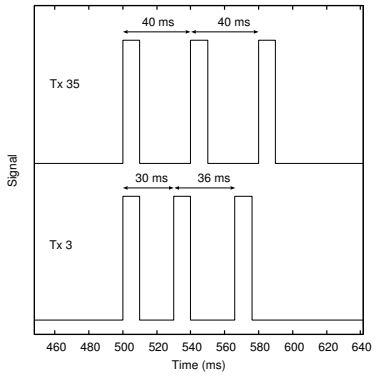


Fig. 5. Example signal timing for two tags

This allows us to remove noisy transmissions more reliably, as we can discount any ultrasound transmission that does not fall within the limits of the expected timing pattern. It also allows for a small amount of error correction; if the timing is correct but positioning is poor for one of the three pulses, then we can remove the outlier.

Tag ID	Period multiplier	$T_1 - T_0$ (ms)	$T_2 - T_1$ (ms)
0	947	30	30
1	953	30	32
2	967	30	34
⋮	⋮	⋮	⋮
35	1193	40	40

Table 1. Tag timing signatures

A further consideration with multiple tags is the use of the shared 40kHz audio channel. In order to reliably identify an individual tag, the tag must have exclusive use of channel for the duration of its three pulses, and for a brief time before and after transmission. It must then remain silent while other tags are active. With this constraint, there is a trade-off between the number of objects that can be tracked and the frequency with which their position can be updated. Thus the object tracker is best suited to applications where a large number of objects are tracked several times per minute, or where one or two objects are to be tracked several times per second.

As the tags are independent of one another, is is not possible to co-ordinate them to transmit consistently in a particular order. However, we reduce the

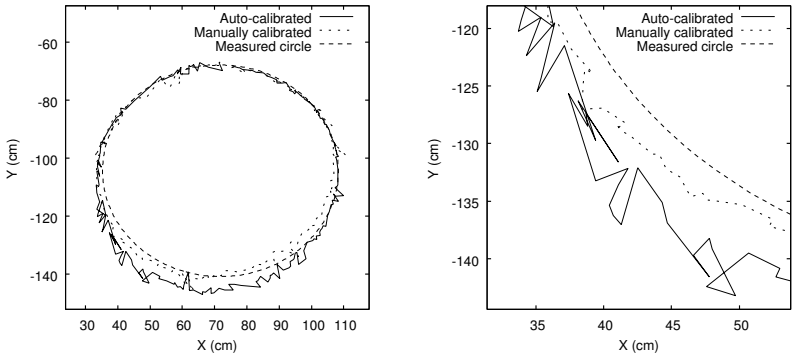


Fig. 6. Moving transmitter track (left) and close-up (right)

likelihood of a collision between one or more tags by configuring each tag to pause for a different length of time after transmitting a set of pulses. By carefully choosing these times to be a constant multiple of the set of prime numbers, we can avoid aliasing, where two tags collide regularly. Table 1 shows possible signal timing patterns for a set of transmitting tags. Note that the prime period multipliers are approximate, since each tag has its own clock which will drift with respect to other tag clocks.

Performance The auto-calibration algorithm was executed using a set of recorded distances as described in Sect. 3. Execution time for calibration was approximately 5 minutes on a 1GHz Pentium processor. The best solution was then used to track a transmitting tag. The computed track was compared with a manually measured ground truth, plus a track computed using manually calibrated receivers. Figure 6 shows an aerial view of these three tracks, overlaid to give an indication of the scale of their accuracies. The close-up view shows the region of greatest difference between the tracks. The slightly lower accuracy of the auto-calibrated result is clear, with computed positions jumping around more erratically than in the manually calibrated track.

We observe a systematic scaling error, resulting in an error of 5-7 cm in the bottom part of the track. This may have been caused by a consistent offset in the distance data recorded. It is important to note that the systematic errors will drop out of the system if it is trained with information about the positions of objects in a room. Often we are only interested in knowing that we are next to some object, rather than precise co-ordinates for measuring distances.

The reproducibility of results using the auto-calibrated system was analysed using sets of over 1000 readings taken at stationary transmission positions. Running this data through both a manually calibrated and an auto-calibrated tracker revealed some variability in the placing of points even after systematic errors were taken into account. These radial errors give a better indication of how reproducible results are using the tracking system. Table 2 shows accuracies

	SEP 50%	SEP 95%	RMS
Manually calibrated system	1.28	2.71	1.70
Auto-calibrated system	1.87	4.02	2.29
Ratio manual/auto	0.68	0.68	0.74

Table 2. 50% and 95% SEPs and RMS errors given in centimetres

of the system in terms of 50% and 95% spherical error probability (SEP) and root mean square error (RMS).

The SEP and RMS values all show that on average the manually calibrated system performs a little better than the auto-calibrated one, though not significantly so. The errors for the auto-calibrated system are still well within the limits of many applications that currently use manually calibrated systems, and the ratios indicate that the distributions of errors are similar in both cases. RMS values give a more representative indication of the system's consistency at producing results.

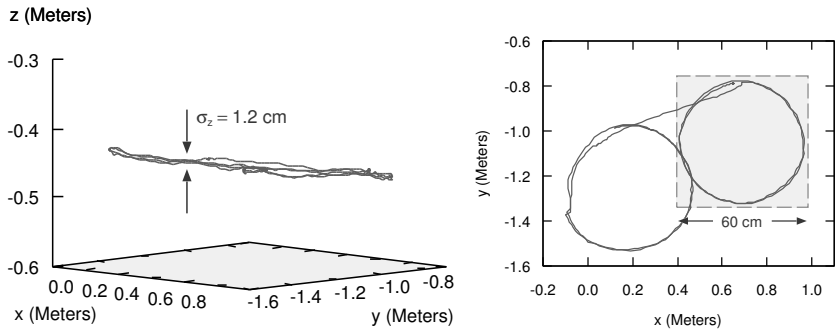
5.2 Single Object

The system can also change modes to track a single object to a high degree of accuracy. By giving one mobile device a monopoly on the 40kHz audio channel we are able to increase the ultrasound transmission rate to frequencies up to 33Hz. This gives our system a shorter response time and allows us to track objects moving at higher velocities and accelerations. When we receive regular readings from one object, we can use a more effective method to calculate this object's position.

Estimator The estimator we use for this mode of operation is the Kalman filter [12]. Although it is not well suited for the large transmission periods of the multiple object tracker, the Kalman filter does fit well with the new tracking conditions. Specifically, it is more efficient and predictable than Levenberg-Marquardt, it uses knowledge of the previous system state, and it provides a method for modelling the dynamics of the system. In our filter, we employ a position-velocity model that assumes the velocity of the tracked object is constant and subject to acceleration 'noise'. In the following process equations, $\boldsymbol{U}$ and $\boldsymbol{V}$ are the position and velocity of the tag, P is the transmission period and E is the transmission time.

$$\begin{aligned}
\boldsymbol{U}_k &= \boldsymbol{U}_{k-1} + P_{k-1} \cdot \boldsymbol{V}_{k-1} \\
\boldsymbol{V}_k &= \boldsymbol{V}_{k-1} \\
E_k &= E_{k-1} + P_{k-1} \\
P_k &= P_{k-1}
\end{aligned}$$

The state vector falls from these equations as $[\ \boldsymbol{U}\ \boldsymbol{V}\ E\ P\ ]^T$.



(a) (b)

Fig. 7. Path of single tag operating at 33Hz, (a) 3D view showing position along z-axis, (b) 2D view showing position along x and y axes.

For each receiver i, we have one measurement equation that relates the state, the known position of the receiver, $\boldsymbol{R}_i$ (from auto-calibration), and the speed of sound, v, to the reception time of the signal, T_i.

$$T_i = E + \frac{\left|\boldsymbol{U} - \boldsymbol{R}_i\right|}{v} \tag{11}$$

Although the reception times recorded by each receiver are obtained almost instantaneously, we integrate the measurements with the filter in a serial fashion. This method, known as single-constraint-at-a-time (SCAAT) Kalman filtering [13], provides advantages in terms of efficiency and outlier detection.

Performance Figure 7 shows the path of a tag mounted on the back of an office chair. The chair was spun on its base, rolled, and spun again to create the two circles illustrated. The height of the tag was kept constant throughout as is reflected by the accuracy of the filter along the z-axis. In terms of standard deviation from the mean position, this value is 1.2cm. To gauge the dynamic accuracy in the x-y plane, we compare the diameter of the circles in Fig. 7 to the true 60cm swivel diameter of the chair. The circles are approximately 6% smaller than the diameter created by the chair. We attribute the discrepancy to errors in the value used for the speed of sound, a non-ideal arrangement of the receivers having a high dilution of precision, varying transmitter-receiver incident angles, and errors in the positions of the receivers from auto-calibration. The stationary 3D accuracy of the system, in terms of spherical error probability over a 40 second time frame, is 5mm for 95% of the readings and 6mm for 99%.

With these accuracies, the single object tracking system could be used with applications such as home gaming systems (Xbox, PlayStation, etc.) and motion tracking for animation. Fused with inertial sensors, the system also has the potential to be used with augmented or virtual reality applications. A combined

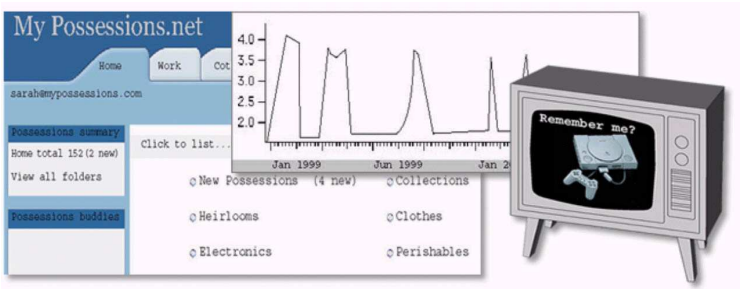


Fig. 8. Initial design sketches for an application that tracks possessions, shows usage over time and re-advertises "ignored" objects using channels in the home.

ultrasonics-inertial-sensor approach has been used successfully by InterSense Inc. in a number of their high performance products [14–16].

6 Explorations of Ultrasonic Tracking within the Home Environment

We have been developing a series of interactive scenarios based around tracking objects within the home. The scenarios follow earlier studies involving domestic probes and experiments within the home [17, 18]. In this environment we want to experiment with designs without having to repeatedly calibrate a tracking system by hand. This is very much work in progress, but our initial experiments offer interesting insights into the use of such a system, and its practical impact and value for Ubicomp applications.

Below we use the term *engineer* to denote a person who designs ultrasonic systems, and the term *designer* to denote a person who designs ubicomp experiences. A *user* is someone who interacts with the final experience.

6.1 A Motivating Scenario

Our aim is to track a series of objects within the home, in order to indicate current object positions and patterns of usage. Small objects are tagged and registered with an application running on a computer connected to the ultrasonic controller. The application shows a list of registered possessions and allows new items to be added. The designer can specify a name, description and photo to map onto an underlying tag ID. During tracking, 3D co-ordinates of the tags are time-stamped and added to a database.

The database is used to provide a map view of the current position of each object, and a graph (called the "affection radar") that plots the distance each object has moved over time. Whilst many objects achieve high initial attention, the novelty often quickly fades. The affection radar allows the user to reflect on an object's changing patterns of usage. The data can also be used by applications.

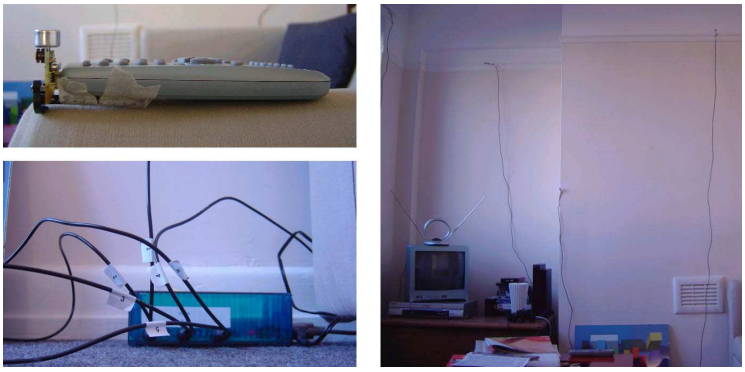


Fig. 9. An initial deployment of the ultrasonic tracker within the home. Tag attached to a remote control (top left). Ultrasonic controller (bottom left) and receivers attached to walls (right).

For example, objects that have become old, ignored or unused could re-advertise themselves to the user through output channels in the home in the form of an occasionally appearing photo. An initial set of design sketches for the application, including the affection radar and re-advertising objects, is shown in Fig. 8.

6.2 Experiments with Ultrasonic Tracking in the Home

In order to realise the scenario described previously, we conducted various trials of the ultrasonic tracker within domestic environments. Key technical questions were: How easy is it to set up? How fine-grained is the tracking in practice? We also need to consider key design questions, such as whether movement is a good metric for determining usage, and how often different types of object move.

Figure 9 shows an installation of the system within the home of a designer. The receivers were fixed to the walls of the space to provide coverage over the living room. Tags were attached to various objects in the home to show movement around the space. These included a chair, remote control, camera, and books. Tag data captured by the receivers were relayed onto a connected laptop, where absolute 3D coordinates were calculated and logged. Initially this recorded data was processed using visualisation software back in the design studio, and only console output was available to verify readings in real-time. After refinement, the visualisation software was also deployed in the home to provide real-time graphical feedback to users.

The system was installed manually at first, and later using auto-calibration to compare experiences. Manual installation involved a designer measuring the co-ordinates of the receivers relative to a point of origin in the middle of the room. Measurements were entered manually into an application running on a laptop. The space contained large household objects (tables, chairs and so on), some of which had to be cleared so that distances could be calculated between

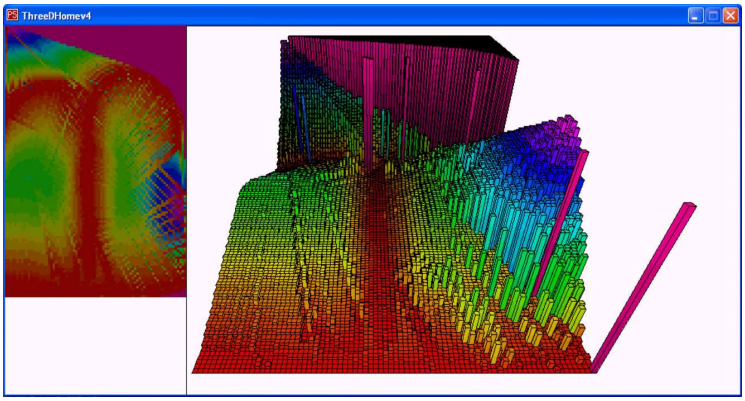


Fig. 10. One example of ultrasonic data visualisations.

the point of origin and the receivers. This added to the initial disruption caused by the placement of receivers.

During tracking, readings for each tag were displayed in the console. The designer used a simple test to verify these readings: manually measure the X, Y and Z distances between corners of a chest of drawers and compare these with distances calculated with a tag. This simple test revealed poor tracking accuracy due to human errors introduced during receiver measurement, and poor coverage of the room. Manual calibration was repeated several times until coverage and accuracy were deemed acceptable. This involved repositioning and recalculation of receiver coordinates, with an engineer to help take measurements. This added considerably to the time taken to calibrate the system.

In contrast, the auto-calibration process took considerably less effort and time. To accomplish this, the designer ran software on the laptop, which captured data from a single tag. The designer moved the tag around the room, taking care to collect distance readings from all parts of the room. After approximately one minute, the designer stopped data capture on the laptop and ran the auto-calibration software. Within approximately 5 minutes a solution had been found, and transmitting tags were tracked in absolute 3D co-ordinates. Various activities were performed to test the tracking capabilities of the system. For example, moving slowly around a dining table or moving from one corner of the room to the centre, and viewing whether the visualisations plotted a similar path. The designer also used a tape measure to manually resolve the 3D coordinates of a tag, testing these against the data generated on the laptop. These tests showed promising results, indicating reliable tracking accuracy of 7cm. After some minor tweaking of receiver positions, rerunning of auto-calibration software, and retesting (using methods described previously), the designer achieved a tracking accuracy of 3cm.

Spatial information captured from the tracking system was interfaced with a design tool called Processing [19]—a popular high-level programming language

used to rapidly prototype graphical applications. Visualisations were generated based on this data, including maps highlighting an object's trajectory through a space, activity maps indicating areas in which objects predominately resided, and maps showing areas in which line of sight to one or more receivers was occluded. For example, the occlusion map in Fig. 10 shows a view of the home where line of sight was occluded by large objects such as a dining table or a sofa (shown in light grey). They also illustrated whether objects did indeed move frequently, and where in the space they were most often used.

6.3 Initial Reflections

Initial trials of the ultrasonic system within a number of domestic settings showed promising results and allowed us to refine our application designs further. During our trials, we found several advantages to using auto-calibration over a manual approach:

- *The ability to quickly reposition receivers and retest in order to maximise coverage and tracking accuracy within the space.* The accuracy and coverage of ultrasonic trackers is clearly heavily dependent on the placement of receivers. We found a need to reconfigure receiver positions several times in order to improve tracking and coverage. This process is greatly simplified if the designer can avoid re-measuring receiver distances each time manually.
- *The ability to minimise human errors being introduced by manually measuring receiver coordinates.* We also found that errors crept in during manual calibration, requiring designers to re-measure receiver positions, a process that would often become frustrating.
- *The ability for a single non-technical designer to setup the tracker.* In our experience, manual measurements require the work of two people, and often involve reaching awkwardly and noting down large numbers of distances and offsets, and carrying out calculations. Non-technical designers can find this process daunting. Auto-calibration obviates the need for this inconvenient process.

During these initial tests, we found that the auto-calibration algorithm works well for most arrangements of receivers. Tracking was most accurate when the receivers were set up on the ceiling and walls to avoid reduced precision in the Z-axis. Care was also taken to point the receivers toward the centre of the tracking area, to maximise the chance of receivers detecting tag transmissions.

Our visualisations indicated that smaller objects such as books, electronic items and toys move much more than larger objects such furniture, as expected. Often patterns arise over a period when viewing the paths of objects through a space—for example, in our deployment the remote control often followed a set path from the shelf by the CD player to the coffee table and back. Therefore one must be careful in using movement as a metric for determining usage, as this only applies to certain objects which are small enough to be mobile and their everyday interaction relies on movement (such as a camera).

Initially, when the system was deployed it was passively capturing data to be processed asynchronously back in the design studio. At this early stage, participants felt that these tests were a little intrusive, voicing concerns about privacy and the ways that data would be used. This level of intrusiveness began to fade once the users were provided with more feedback and control over the data capture. For example, in the second series of tests visualisation software ran on the home desktop, and could be stopped and restarted by the user at any point. Participants felt much more comfortable living with the system when they could see the information captured in real-time, and could choose when it was appropriate for the system to begin capture.

These initial experiments also showed that during everyday use, certain objects would move in and out of a room frequently. As an extension to the scenario one could imagine a setup that incorporated several ultrasonic systems, one in each room—this, however, raises issues about setup time and data management across multiple receiver boxes. Designers also suggested whether more coarse grained tracking technologies could be used to compliment the ultrasonics system, particularly for objects that only move short distances.

Our experiments have shown the viability of using an ultrasonic tracking system as a low cost, fine grained mechanism for tracking objects. We are now moving to realise the interactive applications described earlier.

7 Conclusions and Future Work

We have presented an ultrasonic-only tracking system aimed at users beyond the ubiquitous computing community. The system uses algorithms to self-calibrate and self-orient, so that little technical knowledge is required for setup. It gives the user an easy way to calibrate the system and for aligning the system with the user's origin and principal axes (by walking along an imaginary L).

Tracking is performed in one of two alternative modes, providing a choice between tracking multiple near-stationary objects, or one object at a high frequency. The mode can be changed without a need to re-calibrate the receiver infrastructure. Where multiple objects are tracked, each object is preprogrammed with a unique pulse timing signature.

The results show that tracking multiple objects using an auto-calibrated system can be performed to an RMS accuracy of approximately 2.3cm, which is sufficient for our target applications. Single object tracking achieves accuracies of around 3cm in the dynamic case.

We have deployed our system in the home and allowed designers to create an installation. This has provided experience using the system in a more typical setting than the laboratory, and we have described some initial reflections that may be of use to other researchers wishing to implement a tracking system. Future work is planned to further the accessibility of the system to designers, including the use of toolkits that provide a more meaningful interface to the system's outputs.

References

1. A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. In *IEEE Personnel Communications, volume 4 no.5*, pages 42–47, October 1997.
2. C. Randell and H. Muller. Low Cost Indoor Positioning System. In Gregory D. Abowd, editor, *Ubicomp 2001: Ubiquitous Computing*, pages 42–48. Springer-Verlag, Sept 2001.
3. Michael McCarthy and Henk L. Muller. RF Free Ultrasonic Positioning. In *Seventh International Symposium on Wearable Computers*. IEEE Computer Society, October 2003.
4. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *Mobile Computing and Networking*, pages 32–43, August 2000.
5. Ajay Mahajan and Maurice Walworth. 3-D Position Sensing Using the Differences in the Time-of-Flights from a Wave Source to Various Receivers. In *IEEE Transactions on Robotics and Automation*, pages 91–94. IEEE, 2001.
6. J. J. Moré. The Levenberg-Marquardt Algorithm: Implementation and Theory. In G. Watson, editor, *Lecture Notes in Mathematics, v630*, pages 105–116. Springer Verlag, 1978.
7. Lester Ingber. Very Fast Simulated Re-annealing. In *Mathematical Computing Modelling*, pages 967–973, 1989.
8. Brian Gough. *GNU Scientific Library - Nonlinear Least Squares Fitting*, chapter 36. Network Theory Ltd, 2001.
9. Paul Duff and Henk Muller. Autocalibration Algorithm for Ultrasonic Location Systems. In *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*, pages 62–68. IEEE Computer Society, October 2003.
10. I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
11. L. Ingber. Adaptive Simulated Annealing (ASA). http://www.ingber.com/.
12. R. E. Kalman. A New Approach to Linear Filtering and Prediction. In *Journal of Basic Engineering (ASME)*, pages 82(D):35–45, March 1960.
13. Greg Welch and Gary Bishop. SCAAT: Incremental Tracking with Incomplete Information. In *SIGGRAPH 97 Conference Proceedings, Annual Conference Series*, August 1997.
14. Eric Foxlin, Michael Harrington, and George Pfeifer. Constellation: a wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 371–378. ACM Press, 1998.
15. Eric Foxlin, Michael Harrington, and Yury Altshuler. Miniature 6-DOF Inertial System for Tracking HMDs. In *Aerosense 98, Orlando*, April 1998.
16. Intersense Inc. Website. http://www.isense.com/, 2003.
17. W. Gaver, A. Boucher, S. Pennington, and B. Walker. Subjective Approaches to Design for Everyday Life. In *CHI Tutorial, Ft. Lauderdale*. ACM Press, 2003.
18. W. Gaver, A. Dunne, and E. Pacenti. Cultural Probes. In *Interactions Magazine*, volume VI(1), pages 21–29, 1999.
19. Processing. Website. http://www.processing.org/.