Pers Ubiquit Comput (2007) 11:439–449
DOI 10.1007/s00779-006-0096-1

ORIGINAL ARTICLE

Improving the accuracy of ultrasound-based localisation systems

Hubert Piontek · Matthias Seyffer · Jörg Kaiser

Received: 11 August 2005 / Accepted: 9 November 2005 / Published online: 5 December 2006

Abstract We present an improvement to ultrasound-based indoor location systems like Cricket (Priyantha et al. The Cricket location-support system, 2000). By encoding and modulating the ultrasound pulses, we are able to achieve greater accuracy in distance measurements. Besides improving the distance measurements, we improve the position update rate by synchronising the active beacons. We also propose a method that could further improve the update rate by superimposing encoded ultrasound pulses. Further, an experimental evaluation of our improvements is presented.

Keywords Indoor localisation · Ultrasound · Pulse compression

1 Introduction

Localisation is still one of the challenges in mobile ubiquitous application scenarios. Because knowledge of the position is the basis for location-based services, navigation and ad hoc cooperation, considerable research has been expended and many approaches can be found already as working systems and in the literature. However, most of them only allow a rather coarse grained determination of the position. Consider the coordination of autonomous vehicles which uses accurate location to coordinate access to junctions or intersections on a factory floor or in a large stock building. It would be highly beneficial if this could be based on an accurate and reliable positioning system. Another example that we aim at is to use robots in a "mixed reality" scenario where they move in a simulated virtual scene and interact with virtual robots. This application requires a very accurate localisation of the robots in the real world. Applications like this add a new dimension to techniques of simulating real-world settings. We consider in our application mobile robots with a physical size of about 0.4 m in length which move with a moderate speed of about 0.7 m s^{-1} (see Fig. 1). From this, we can derive some primary requirements for the location system: the accuracy of the system must at least be in the order of the size of the robots or better. Our design goal here is a position accuracy of 0.30 m. The position update rate must be at least 1 s^{-1}. Our robots are capable of moving up to 0.7 m s^{-1}. This would correspond to about two robot lengths at full speed. The system must be scalable in space and in the number of clients. A crucial point in a localisation system often is the question of the division of labour between the infrastructure provided in the environment and the respective components on the mobile vehicles. This also affects problems of energy consumption and required computational resources. In a mobile robot, a substantial amount of energy is needed for mobility of vehicles. Compared to this, the energy need for the moderate computational requirements of a location system may be relatively low. However, when considering the infrastructure, it

H. Piontek (✉) · M. Seyffer ·
Universität Ulm, Albert-Einstein-Allee 11, 89069 Ulm, Germany
e-mail: hubert.piontek@uni-ulm.de

M. Seyffer
e-mail: matthias.seyffer@informatik.uni-ulm.de

J. Kaiser
Institut für verteilte Systeme,
Otto-von-Guericke-Universität Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany
e-mail: kaiser@ivs.cs.uni-magdeburg.de

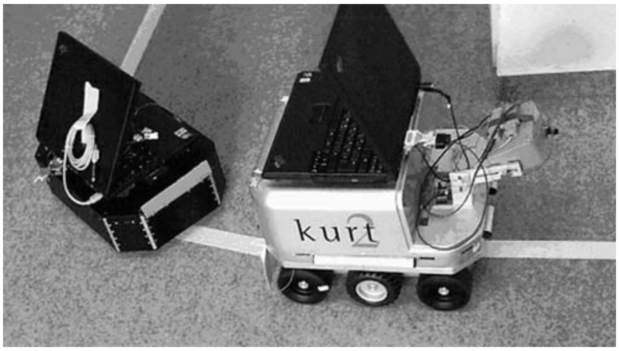


Fig. 1 Co-operating robots

should be simple, easy to deploy and, because it often needs to be operated with a local power source, the energy demand should be low. Thus, we aimed at a relatively simple infrastructure composed of simple beacons with low computational requirements and low energy consumption. The principle of operation is based on distance measurements to at least three beacons and subsequent trilateration. The distance is determined by the differences between the time which a radio signal and an ultrasound signal need to travel from a beacon to the respective receivers on the mobile entity. This is very similar to the principle of the Cricket location system [1]. Actually, we first built the hardware of a Cricket system and tried to use it in our mobile environment. However, we discovered two major drawbacks of this system:

1. It was rather difficult to obtain a precise edge of an ultrasound signal. This is mainly because of the limited bandwidth of the ceramic ultrasound transducer. Even when we used a high-energy ultrasound pulse, which is not desirable because of energy constraint, the results were not satisfactory. It limits the localisation accuracy substantially.
2. The update rate of the Cricket system is not sufficient to accommodate high accuracy localisation of mobile systems.

The contribution of this paper therefore is to investigate ways of how to improve such a location system. Firstly, we will present and evaluate an impulse compression technique to overcome the first drawback. Then we will describe ways to improve the update rate. The rest of the paper is organised as follows: Sect. 2 discusses related work. Section 3 will introduce the general architecture of the location system and will describe the hardware. It also will briefly address the synchronisation of the beacons to avoid collisions on the radio channel. Section 4 will sketch the pulse compression techniques and present results. Section 5 will give a short overview over the activities to improve the update rate of the location system. Finally, we will provide a conclusion.

2 Related work

There are three basic methods to determine one's location:

1. Sensing the location by means of a *sensor grid*: a sensor (e.g. magnetic or pressure sensitive) can detect an object in its close vicinity. Distributing those sensors in a regular grid allows determining one's position within this grid. The most common example for such a system is a touch screen as found on most PDAs. For our application scenarios of autonomous robots, such (tight) sensor grids would be too costly to deploy. The original Cricket [1], the Active Badge Location System [2] and 3D Locus [3] are of this kind. Both provide location information at the granularity of a room, which does not meet our requirements.
2. Sensing the direction of at least two landmarks (of known location) to a common reference (for 2D positioning), and using *triangulation* to determine one's position.
3. Sensing the distance to at least three landmarks, and using *trilateration* to determine one's position. A popular example for this method is GPS, which is proven to work reliably at accuracy down to several meters in an outdoor environment. However, it does not work inside buildings. Laser technology is known for its accuracy, but we considered it to be too expensive, and too complex for our goals. The Cricket Compass [4], Cricket v2 [5], The Bat [6, 7] and RADAR [8, 9] are examples for this type of positioning systems that work indoors. Because we are aiming at a distributed system that is composed of fixed, active beacons and passive mobile clients, The Bat and RADAR are ruled out. The Cricket system comes closest to our goals: it uses easily deployable components. The autonomous system architecture fits nicely into our view of the world where we aim at autonomous components that interact without any central coordination.

RADAR [8, 9] is an RF-based system for locating (and tracking) users inside buildings. It uses the signal strength information from wireless networking equipment. The system is capable of locating users to within a few meters of their actual location. The system uses a combination of empirically determined and theoretically computed signal strength information, as the propagation of RF inside buildings is hard to cope with. One of the advantages of RADAR is that the means to provide location information also provides traditional data network services. However, the traceable entities are laptop computers. A major drawback is the need for empirical data, which must be collected before the system can go to *real-time mode*, i.e. into operation.

Cricket [1] originally aimed at supporting a user to find his location within 1 or 2 sq ft. The system architecture is similar to our own: Cricket uses *beacons* that basically advertise their position. *Listeners* that move through instrumented areas use the time of flight of ultrasound signals to estimate the distance to all beacons in (ultrasound) range. The current position is the area advertised by the beacon which is closest to the listener. Cricket's *granularity* is *portions of a room*. As in any other system utilising both radio and ultrasound signals, corresponding radio und ultrasound signals must be correlated at the receiver. Cricket does not modulate the ultrasound signal, so it needed a different mechanism: typically, radio signals can be received at much greater distances than ultrasound signals. This ensures that whenever an ultrasound signal is received, so is the radio signal. Using a small bandwidth radio link, and having long enough radio messages, it is assured that the ultrasound signals arrive while the radio message is still being transmitted. In the absence of interference, this ensures that the correct correlation of radio and ultrasound signal is done. Errors in measurement due to changes in the speed of sound, e.g. due to temperature, are irrelevant because only the closest beacon is used to determine the current position.

The Cricket Compass [4] improves the original Cricket system on the listener side only. Besides mere distance measurement, the receiver orientation towards the beacon is determined. This is done using five ultrasound receivers in a V-shape, and measuring phase differences in the incoming ultrasound signal. With Cricket Compass, positioning in terms of absolute coordinates within a room was introduced. This requires at least four beacons. The method described in [4] overcomes the problem of not knowing the speed of sound, so no further sensory equipment is needed.

Cricket v2 [5] is based on improved and simplified hardware components. Cricket hardware units can be configured to either be a beacon or a listener. The API has been extended, and a software distribution allowing developing Cricket enabled applications, e.g. in Java is available.

With their 3D-Locus system for archaeological excavations [3], the LOPSI research group developed an ultrasonic positioning system that allows positioning with an accuracy of 5 mm. The system uses long BPSK modulated pseudo-noise ultrasound signals and radio transmissions for synchronisation. The LOPSI group uses broadband ultrasound transducers with a bandwidth of 15 kHz. 3D-Locus is organised as a centralised system with active beacons to be located and a static setup of receivers. To cope with wind speed and air temperature, a reference transmitter at a known position is used. While the active beacons are rather low-cost devices incorporating only a small microcontroller, the receivers are more sophisticated and use DSPs that are connected to a workstation that does the actual positioning. Ultrasound transmissions are done in pairs of beacons at a time. Scheduling is done by the central control system.

Hazas and Ward [10] describe a new custom made ultrasound transducer unit they call *Dolphin*, and its usage in a centralised system. The Dolphin units have a broad usable bandwidth of 76 kHz. Hazas and Ward [11] describe a modified version of their positioning system where the moving objects calculate their positions and the infrastructure is composed of active beacons. Even though the Dolphin units seem to be predetermined for our application, this is not the case at the moment. They are too bulky, consume too much energy, and most importantly they currently are too expensive to manufacture.

A more comprehensive overview [12] is available from the LOPSI research group.

3 Positioning system

The basis for our location system is distance measurement. A *client* measures its distance to several (three or more) *beacons*. Taking these distance measurements, it calculates its position using multilateration. Beacons are attached to the ceiling. Both beacons and clients are equipped with radio modules and ultrasound transducers. Within each room, all beacons should be mounted at the same height (see Fig. 2a). A client (e.g. on a mobile robot) can determine its position in a three-dimensional space by measuring the distance to at least three beacons: two measurements limit one's position to somewhere on a circle (denoted by the thick vertical black line between the intersections of the two circles in Fig. 2b). The third measurement reduces this to two points on this circle (the intersection of both circles in Fig. 2c). One of them can be discarded, as it is above the beacons, which is impossible because of the directional characteristics of the ultrasound transmitters. The measurements ideally

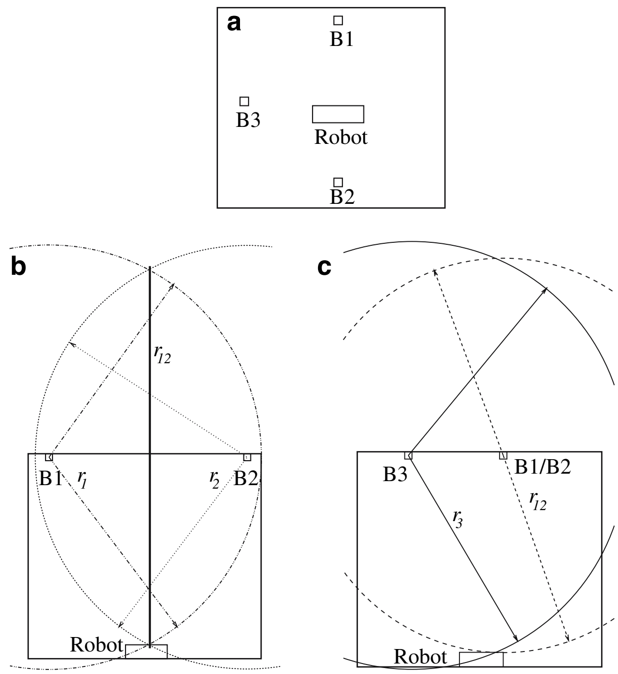


Fig. 2 **a** Room seen from above (*X*–*Y* plane). **b** Room seen from the side (*Y*–*Z* plane). **c** Room seen from the side (*X*–*Z* plane)

should be done simultaneously, especially if the robot is moving.

3.1 Distance measurement

Distance is measured using the difference in time-of-flight of RF signals and ultrasound signals. The time difference for travelling a distance *d* between the ultrasound signal and the radio signal is

$$t = t_{us} - t_{rf} = \frac{d}{v_{ultrasound}} - \frac{d}{v_{radio}}.$$

For a distance *d* of 10 m, the radio signal needs about $t_{rf} \approx 30$ ns. The ultrasound signal, however, will need about $t_{uf} \approx 30$ ms. As $t_{rf} << t_{us}$, t_{rf} can safely be omitted from the above term. Unfortunately, the speed of sound is not constant. Indoors it varies mainly with temperature. Between –20 and 40°C, it can be approximated in a linear fashion: $v_{ultrasound} = (331.6 + 0.6T) m/s$ where T is in °C. Not dealing with temperature would introduce rather large errors, e.g. 3.4% or 0.34 m when measuring a distance of 10 m and going from 10 to 30°C. There are two possibilities for dealing with unknown speed of sound: (1) try to approximate the speed of sound using sensors, e.g. temperature sensors, and (2) using one more beacon, and introduce it as another unknown variable in the positioning calculations. Of course, the latter method assumes that the speed of sound remains constant within a room for the duration of a positioning operation. This enhancement is based on the work of the Cricket Compass [4].

3.2 Synchronisation of beacons

The beacons become active in a time-triggered fashion, both for simplicity, and for avoiding collisions (see Sect. 5). The timely properties of the radio channel are well known in our setup, as we use low-power RF modems (LPRS Easy-Radio ER400TRS, see http://www.lprs.co.uk) that show a well-defined behaviour.

3.3 Hardware

Our initial approach was to build a Cricket v1 clone using the same analogue tone decoder for detecting the ultrasound pulse that was originally used with Cricket. The beacons simply sent a constant ultrasound "tone" for a short period of time. The receiver should ideally detect this tone right at its first "edge". In practice, it took the tone decoder several milliseconds to detect the incoming carrier tone. With this setup, the distance measurements had errors in the range of several tens of centimetres for perfectly aligned ultrasound transmitters and receivers. When the ultrasound parts were only slightly misaligned, we had even worse readings (see also [13]). We were able to improve the system by discarding the analogue tone decoder. Instead, we fed the amplified input signal to a comparator circuit. The output is a binary signal that was directly fed into a microcontroller's capture unit. Tone detection was done in software [14]. We recently became aware that Cricket changed to the same technique [13, 15]. The results were promising for aligned ultrasound transmitters/receivers: all measurements were within ±2 cm of the actual distance. Measurement errors grew with the misalignment of the transceivers. When misaligned by more than 35°, the measurements became completely unreliable. Our current approach is to use pulse compression on the ultrasound channel to get accurate distance measurements. The theoretical background is briefly described in Sect. 4 and in more detail in [16]. The beacons (see Fig. 3a) use an 8-bit microcontroller (Freescale 68HC908). They communicate via radio modules. The modulated ultrasound pulse is created in software, so the hardware is kept as simple as possible. The clients (see Fig. 3b) are more sophisticated. The incoming ultrasound signal is amplified and fed directly to a low-power DSP (Freescale 56F800, about 30 MIPS). The DSP is in charge of demodulating and correlating the incoming signal. It essentially sends a

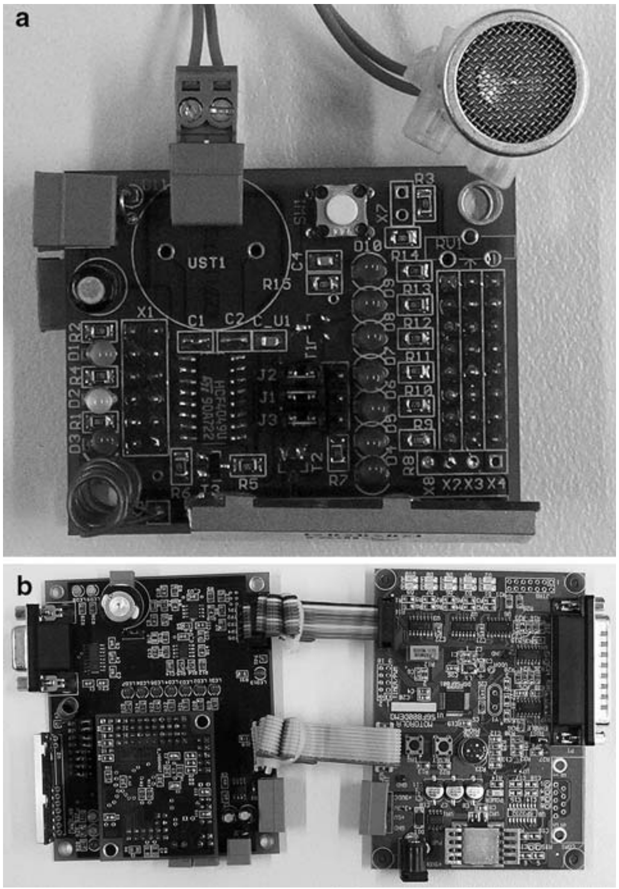


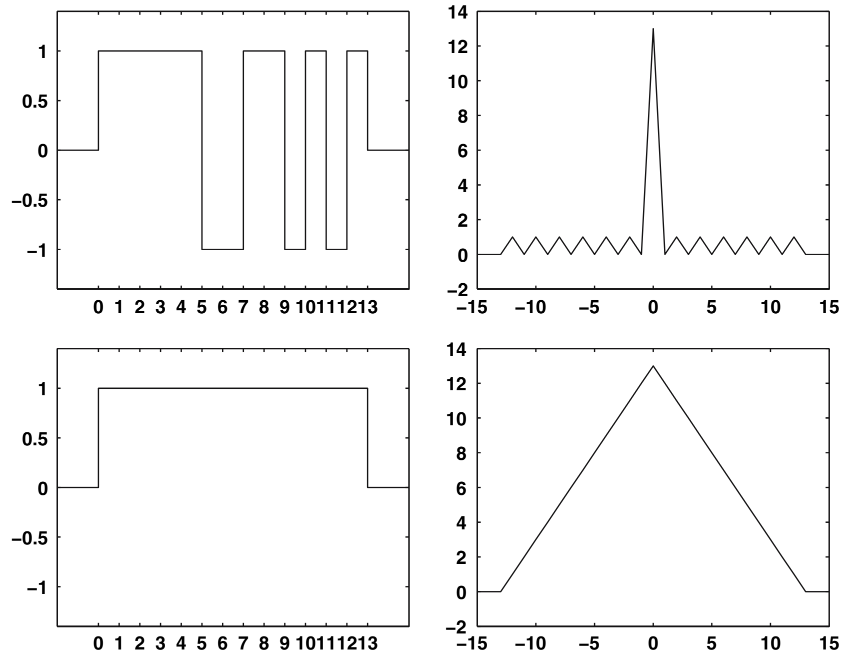
Fig. 3 **a** Beacon. **b** Client

timestamp to an 8-bit microcontroller (same type as on the beacons) which then does the final calculation of the distance to be measured.

4 Pulse compression

The resolution of distance measurements is directly proportional to the sending of an extremely short pulse (a *Dirac pulse*). Sending a very short, single pulse, however, leads to misdetections, as random noise could be misinterpreted as the expected single pulse. There are two solutions to misdetections: (1) make the pulse a very high power pulse, so that its signal-to-noise ratio is good enough, or (2) generate a longer and weaker pulse with the same amount of energy as the short and very high power pulse. Due to the limited bandwidth of approx. 2–4 kHz of ceramic ultrasound capsules and their voltage limit of about 15 V, it is impossible to generate a short pulse with high enough energy. The drawback of the longer and weaker pulse is worse distance resolution, but it can be sent with narrow band devices. To achieve proper distance resolution, the ultrasound signal must be encoded in an appropriate way, called *pulse compression*. The receiver can then apply correlation filters that transform this long, low-power pulse into a short peak that gives a similar distance resolution as the short, high-power pulse. To encode the signal several methods can be used. But considering the computational power of our beacons, we had to use a simple method. We encode our signal using binary pseudo-noise sequences (PN sequences). These are modulated using binary phase shift keying (BPSK). BPSK matches the computational abilities of our beacons and achieves a good coding efficiency of 1 baud Hz^{-1} of bandwidth. For our ultrasound transducers, this yields a maximum data rate of about 2,000 chips per second (as we have a usable bandwidth of about 2 kHz). The PN sequences must have the characteristic to provide a good autocorrelation function to get a sharp peak. Barker codes are a class of well-known codes that possess the required correlation properties (see Fig. 4). The disadvantage of Barker codes is the limited maximum code length of 13 chips. The Barker code's auto-correlation exhibits a sharp edge in contrast to the ''triangle'' shape of a simple ping's auto-correlation. This shows that a ping, as used in our first experiments (and in Cricket), is not very suitable to achieve good distance resolution. Demodulation and correlation on the receiver side are done in software on the DSP. For best results the signal must be sampled at a rate of 160 kHz. The optimal receiver must correlate the incoming signal with the stored reference signal continuously. However, such a receiver would consume about 166 MIPS at a rate of 2,000 chips per second. We use a modified BPSK demodulator (see Fig. 5) to limit the demand for computational power. First, the received signal is transformed from a pass band signal to a base band signal by the quadrature mixer. The signal is split into an in-phase and a quadrature component. Before a data reduction stage, the signals are low-pass filtered. The resulting signals are correlated with the stored Barker code using the schema in Fig. 6. These modifications to the BPSK demodulator reduce the computational requirements to about 10 MIPS. Of course, these modifications degrade accuracy. The overall accuracy we experienced is well within our requirements (see Sect. 4.2). The resulting signal of the receiver is an envelope, showing how good the received signal matches the stored signal. The best match and thus the time of arrival of the signal can be easily determined by a search for the global maximum.

Fig. 4 BPSK modulated Barker code (*top left*, 13 chips), and its auto-correlation (*top right*) versus ping (*bottom left*, 13 chips), and its auto-correlation (*bottom right*)



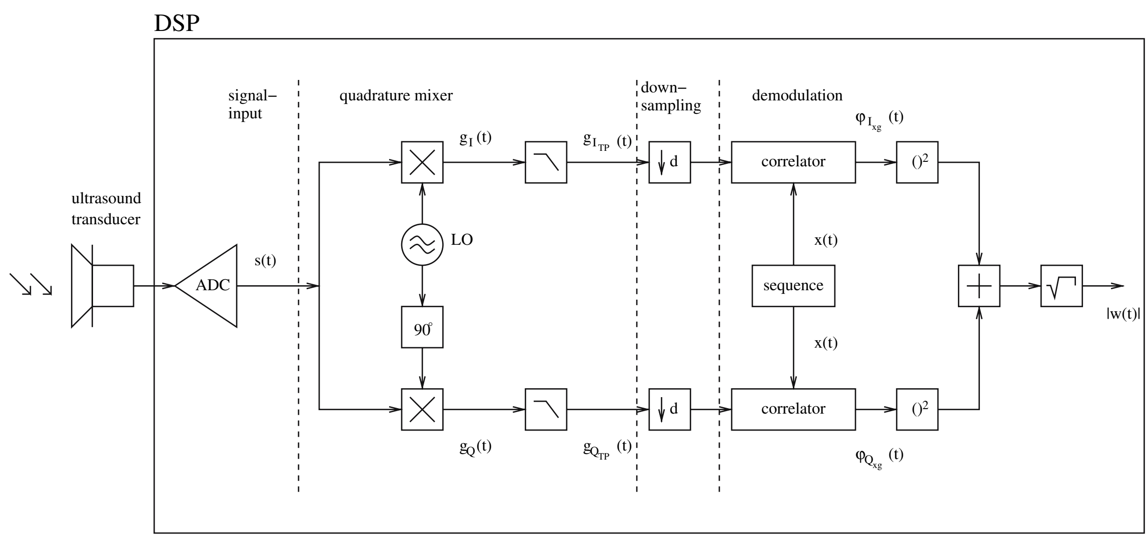
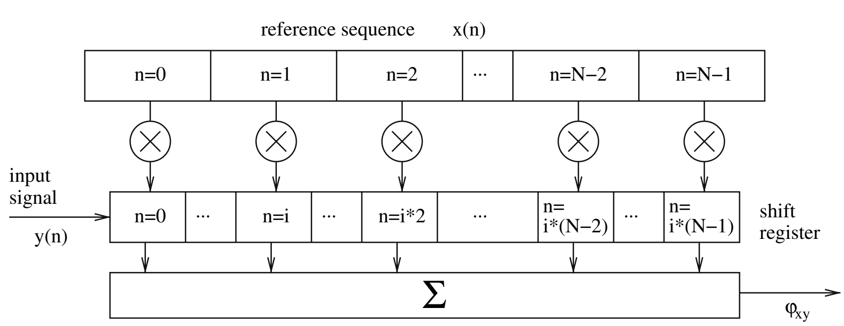


Fig. 5 Demodulation of the incoming signal

Fig. 6 Correlator



4.1 Theoretical results

Using a sampling rate of 160 kHz, a theoretical best-case resolution of about 2 mm could be expected. However as the signal is phase coded, the resolution cannot be less than the wavelength of the ultrasound signal, which is about 8.6 mm. To achieve this accuracy in practice, very long PN sequences are needed, which would affect memory and CPU needs as well as the position update rate. The chip rate of the signal gives an absolute worst-case resolution (upper bound) of 20 cm.

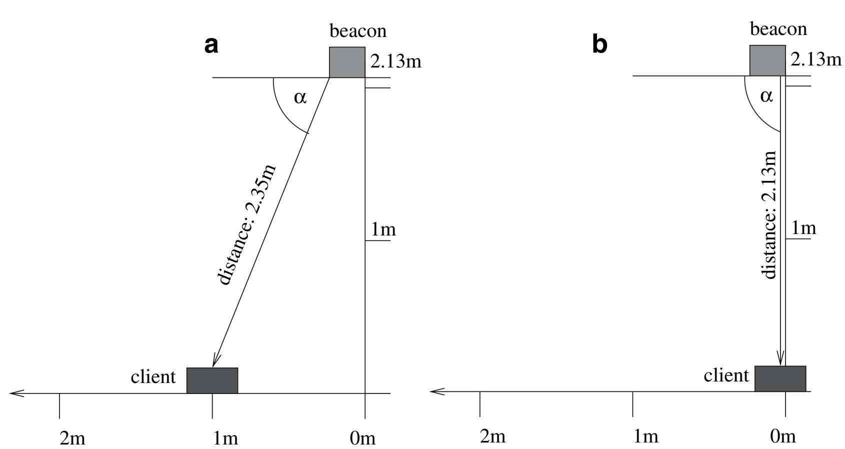
4.2 Experimental results

To get an idea of the behaviour in the real world, the test setup in Fig. 7 was used for first experiments. The data rate was set to approximately 1,666 chips per second and the 13-chip Barker code was used. Several measurements were done at distances from 0 to 2 m. The sender was mounted below the ceiling 2.13 m above the receivers. The first test showed the expected result. It is presented in Fig. 8a. The upper graph shows the received ultrasound signal. The BPSK encoded Barker code is visible starting at a time of about 7 ms, corresponding to a distance of 2.35 m. An echo is visible at approximately 19 ms, corresponding to about 6.5 m. The lower graph shows the ''detection envelope'' with the global maximum corresponding to the beginning of the incoming waveform, and a local maximum for the weak echo. The second test (Fig. 8b) shows unexpected behaviour. The signal is superimposed by several reflections and echoes. In the lower graph, it can clearly be seen (by a human observer) that the beginning of the signal was correctly detected at approximately 6 ms, or 2.13 m (the first major peak of the detection envelope). However, the global maximum results from a reflection. This is possible because of the direction dependent attenuation characteristics of the ultrasound transducers. This problem shows that a search for the global maximum cannot be used in practice. We are working on a heuristic method to find the first peak in the envelope. Leaving out these misdetections, the overall accuracy of the system is within 10 cm. This may not look like an improvement over the comparator-based approach, however the error bound of 10 cm holds for all correctly detected ultrasound pulses. Misaligned transceivers do not lead to growing errors in distance measurements.

5 Improving the position update rate

The most obvious way to improve the position update rate compared to Cricket is to synchronise the beacons among each other. We chose a simple TDMA scheme combined with the separation of larger areas into cells. A cell usually corresponds to a room, because the walls of a room create a natural boundary for the ultrasound waves. A cell is assigned to a single time slot, which is further divided into sub-time slots. Inside a cell, a beacon is assigned to a single sub-time slot. All time slots have the same predefined length and consist of the same number of sub-time slots. The beacons listen to the radio channel to synchronise to the beginnings of their sub-time slot inside the time slot of their cell and to send the ultrasound pulse within its assigned sub-time slot. This avoids collisions on the radio channel among different cells and also among nodes within the same cell, as well as collisions on the ultrasound channel inside the cells. Offline network planning and scheduling helps to improve the update rate: unlike ultrasonic waves, radio waves penetrate walls, but limiting the transmission power of the RF devices also limits the range of the RF signals. The network layout can be chosen so that it is possible for two or more cells

Fig. 7 **a** Test 1. **b** Test 2



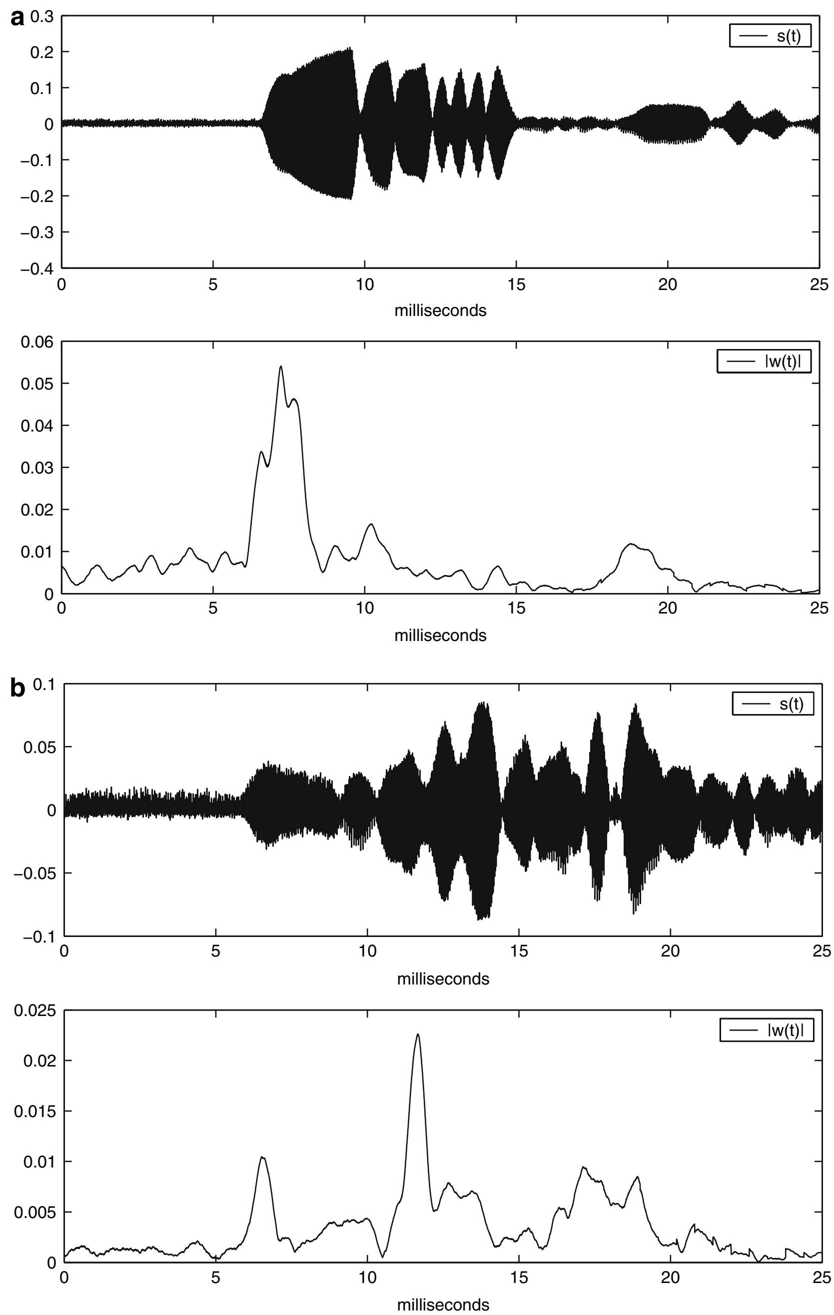


Fig. 8 **a** Test 1: input gain at factor 100, horizontal distance 1 m. **b** Test 2: input gain at factor 100, horizontal distance 0 m

to transmit at the same time without interfering with each other. This reduces the time until a cell becomes active again, and therefore decreases the update rate less than assigning each room its own time slot. This is illustrated in Figs. 9 and 10. Figure 9 shows a typical office environment, where the values in the brackets mean [room number, time slot number]. The fist cell plan is the simplest one assigning to every room a single time slot. This significantly reduces the update rate compared to a small deployment with only a single cell. Figure 10 shows the corresponding timing chart. The second cell plan only uses three time slots and therefore the update rate is considerably higher.

The synchronisation among cells is still guaranteed, because the beacons can listen both to their own cell and to all immediately neighbouring cells. This allows them to synchronise to each other.

Using a rate of 2,000 chips per second and the 13-chip Barker code, the ultrasound pulse has a duration of 6.5 ms. Within 60 ms this pulse can travel about 20 m. After this distance, it is not detectable anymore. That means we are able to do one distance measurement in

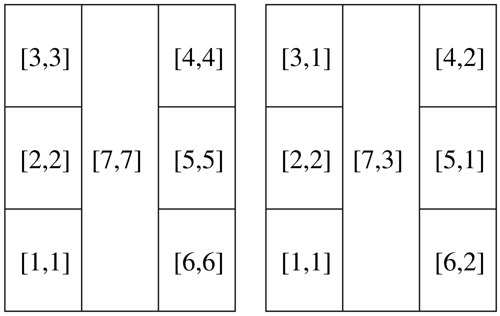


Fig. 9 Cell structure in a typical office building

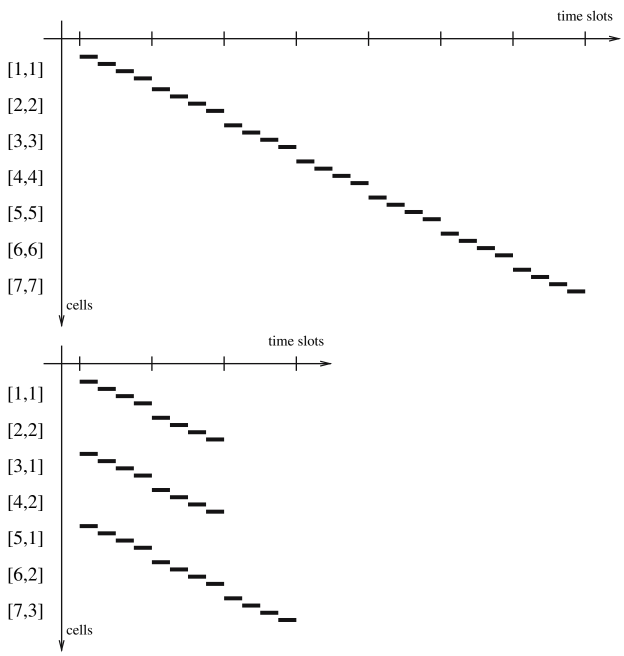


Fig. 10 Timing chart for the cell structure from Fig. 9

66.5 ms, or 15 measurements per second. Because we need three measurements to calculate a position, we can achieve a position update rate of 5 Hz inside a single cell. As mentioned above, this is reduced when using multiple cells. More specifically, we yield an update rate of 5 Hz divided by the number of time slots.

To further improve the position update rate, tests with multiple beacons sending simultaneously were done. To separate signals from different beacons, these signals must be encoded with orthogonal sequences. Because of the speed of sound, the ultrasound pulses normally do not arrive simultaneously at the receiver, so pseudo-orthogonal sequences are needed. Pseudo-noise sequences (PN sequences) have these characteristics. Each beacon uses a different PN sequence. Because there exists only one Barker code for any given code length, we used sequences from the group of *Gold sequences*. Due to the limited data rate only short sequences in the range of 8–16 chips are appropriate. Our experiments show that these sequences are too short to separate the incoming signals. Figure 11 (left) demonstrates this. The first diagram shows the auto-correlation (ACF) of a 16-chip Gold sequence. The second diagram shows the cross-correlation (CCF) of two different 16-chip Gold sequences. They match up to 50%, so the orthogonality of such short sequences is not satisfactory. The third diagram displays the CCF of one Gold sequence with four superpositioned Gold sequences. Because the three other sequences act as noise for the one sequence searched for, the side lobes reduce the processing gain for detecting the sequence. In contrast to the short sequences, Fig. 11 (right) shows the ACF, CCF of two different, and CCF of four superpositioned PN sequences with the original pseudorandom bit sequence with a length of 1,024 chips.

Another handicap results from the transmit characteristics of the ultrasonic transducers. To cover a room, misalignments of the transmitter and receiver must be accepted. These misalignments can lead to differences in received signal power from two different beacons at the receiver by amounts that easily exceed 20 dB. The needed processing gain for a sequence must therefore be even higher and is calculated as Processing gain (dB) = 10 × log(length of Sequence[chips]) (see Table 1). To achieve a high enough probability to successfully separate the sequences, the minimum sequence length should be 256 chips [17, 18]. This will leave a small extra noise margin for successful decoding. Using such code lengths results in the need of significantly more computational power and memory for decoding. Sending 256 chips at a rate of 2,000 chips per second takes 128 ms compared to 6.5 ms for sending a 13-chip Barker code. This method seems to be usable only for ultrasound transducers with a higher bandwidth, and therefore higher possible chip rate. Subsequently, sending a 13-chip Barker code and waiting several milliseconds for it to fade away yields an update rate that is comparable to the method of simultaneously sending long codes. The increased effort in decoding simultaneously sent ultrasound signals does not achieve enough improvement to justify itself.

As there are limits to the achievable position update rate, work must be done to extrapolate the position from previous measurements and fuse this information with other sources of location "hints" like odometry, acceleration sensors or gyros. We believe that this will provide a reliable and up-to-date source of location information.

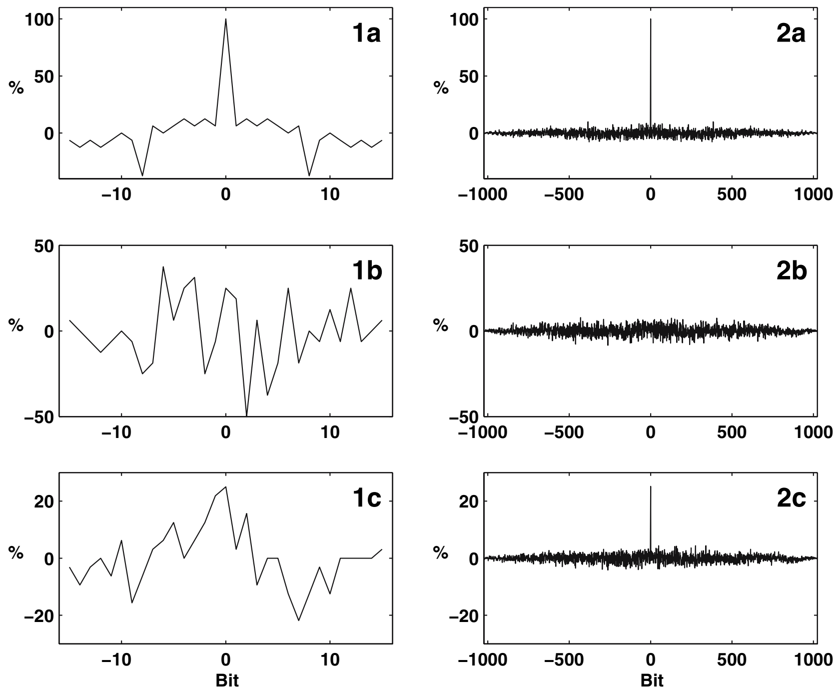


Fig. 11 ACF, CCF of two different, CCF of four superpositioned sequences at lengths of 16 chips (Gold sequences, *left*) and 1,024 chips (PRBS, *right*)

Table 1 PN-sequence length and processing gain

Length of sequence (chips)	Processing gain (dB)
8	9
16	12
32	15
64	18
128	21
256	24
512	27
1024	30

6 Conclusion

In this paper we discussed the problems of a high accuracy localisation system based on distance measurements which exploit the differences in the travel times of ultrasound and radio waves. We showed some intrinsic problems of ultrasound which mainly result from the low bandwidth of the transducers and introduced pulse compression techniques to obtain sufficiently accurate signal detection, crucial for the accuracy of the distance measurement. Secondly, we briefly discussed the problem of improving the position update rate by coordinating beacons and by using orthogonal sequences that allow the ultrasound signals to be sent completely concurrently. The second method turned out to be feasible only with a high computational overhead and also, because the length of the sequences, the benefits are questionable.

References

1. Priyantha B, Chakraborty A, Balakrishnan H (2000) The Cricket location-support system. In: Proceedings of the 6th ACM MOBICOM, Boston, MA
2. Want R, Hopper A, Falcao V, Gibbons J The active badge location system. Olivetti Research Ltd., Cambridge
3. Jiménez AR, Morgado F, Seco F (2004) Ultrasound position estimation sensor for precise localisation of archaeological findings. In: Eurosensors XVIII, Rome, Italy
4. Priyantha B, Miu A, Balakrishnan H, Teller S (2001) The Cricket compass for context-aware mobile applications. In: Proceedings of the 7th ACM MOBICOM, Rome, Italy
5. Cricket Project (2004) Cricket v2 User Manual. MIT Computer Science and Artificial Intelligence Lab, Cambridge, http://www.cricket.csail.mit.edu/v2man.html
6. Addlesee M, Curwen R, Hodges S, Newman J, Steggles P, Ward A, Hopper A (2001) Implementing a sentient computing system. IEEE Comput Mag 34(8):50–56
7. Harter A, Hopper A, Steggles P, Ward A, Webster P (1999) The anatomy of a context-aware application. In: Proceedings of the fifth annual ACM/IEEE international conference on mobile computing and networking, MOBICOM'99, Seattle, Washington, USA
8. Bahl P, Padmanabhan VN (2000) RADAR: an in-building RF-based user location and tracking system. In: Proceedings of the IEEE Infocom 2000, Tel Aviv, Israel, vol 2, pp 775–784
9. Bahl P, Balachandran A, Padmanabhan VN (2000) Enhancements to the RADAR user location and tracking system. Microsoft Research Technical Report
10. Hazas M, Ward A (2002) A novel broadband ultrasonic location system. In: Proceedings of UbiComp 2002: fourth international conference on ubiquitous computing, Göteborg, Sweden
11. Hazas M, Ward A (2003) A high performance privacy-oriented location system. In: Proceedings of the first IEEE

international conference on pervasive computing and communications (PerCom 2003), Dallas-Fort Worth, USA

12. Jiménez AR, Seco F, Ceres R, Calderón L (2004) Absolute localization using active beacons: a survey and IAI-CSIC contributions
13. Balakrishnan H, Baliga R, Curtis D, Goraczko M, Miu A, Priyantha B, Smith A, Steele K, Teller S, Wang K (2003) Lessons from developing and deploying the Cricket Indoor Location System, MIT Computer Science and Artificial Intelligence Lab
14. Uhl K (2003) In-House Lokalisierungssystem zur Positionsbestimmung von Robotern. Lab Report, Universität Ulm
15. Cricket Project: Schematics of a Cricket Mote, MIT Computer Science and Artificial Intelligence Lab, Cambridge, MA, http://www.cricket.csail.mit.edu/software2.0/Cricket_Mote_Schematic_6310-0335-02_b.pdf
16. Seyffer M (2004) Lokalisation in Gebäuden mit intelligenten Sensoren. Diplomarbeit, Universität Ulm
17. Lüke HD (1992) Korrelationssignale, Korrelationsfolgen und Korrelationsarrays in Nachrichten- und Informationstechnik, Messtechnik und Optik. Springer, Berlin Heidelberg New York
18. Skolnik MI (1990) Radar handbook, 2nd ed. McGraw-Hill, New York