Experiences with Cricket/Ultrasound Technology for 3-Dimensional Locationing within an Indoor Smart Environment

Harry S. Sameshima

Edward P. Katz

February 2009

MRC-TR-2009-01

Carnegie Mellon University – Silicon Valley Campus

Experiences with Cricket/Ultrasound Technology for 3-Dimensional Locationing within an Indoor Smart Environment

Abstract

Cricket ultrasound locationing devices are employed to support SmartSpaces, a ubiquitous smart computing research environment at Carnegie Mellon University Silicon Valley. A key requirement for realization of the SmartSpaces vision is a robust (interior) location-reporting system. This paper discusses some early CMU Silicon Valley experiments with Crickets within the context of the SmartSpaces project with an interior setting. Cricket hardware strengths and weaknesses are evaluated and reported in the context of controlling a Roomba robot vacuum. Experiences using Crickets to guide Roomba navigation are discussed.

Introduction

Carnegie Mellon University Silicon Valley's SmartSpaces smart environment research project is an effort to explore a ubiquitous computing platform within the domain of a smart environment intended to be used in human-inhabited living environments, e.g. a home setting. The physical environment is equipped, in part, with sensors, a location reporting system, and a wireless broadband network. SmartObjects within this SmartSpace are capable of collaborating with each other and thus make possible many novel scenarios. A key requirement for realization of the SmartSpaces vision is a robust location-reporting system. This paper discusses CMU Silicon Valley's experiments with Crickets, an ultrasonic locating device, within the context of the SmartSpaces project, and for purposes of demonstration are used to control an iRobot Roomba robot vacuum cleaner.

Background

Modern homes have several computers in the form of special-purpose embedded devices found in such appliances as remote controls, microwave ovens, washing machines, and mp3 players. A growing number of modern homes also have an available local area network (LAN), either wired or wireless (typically Wi-Fi or Bluetooth). Both the embedded computers and the networking technology are becoming cheaper and more commonplace. One interesting deficiency (with minor exceptions) is that most embedded systems don't use the LAN to communicate with each other or with laptops, desktop computers, or mobile phones. Appliances are restricted to doing a few things very well but are unable to combine their activities with other devices in sophisticated ways that would make life easier for humans. Carnegie Mellon Silicon Valley wants to bridge these gaps in the SmartSpaces research project. The SmartSpaces framework can be used for a number of different kinds of inhabitable spaces. In this particular instance, a home setting is chosen, containing one or more enabling wireless communications capabilities (Wi-Fi, Bluetooth, IR, ZigBee, etc.)

The SmartSpaces home setting project envisions sophisticated home appliances that are controlled by software agents that can communicate via a LAN. The appliances are not intelligent, per se. Instead, the combination of a device with a controlling software agent is what makes up a SmartObject. The agent is a wrapper that abstracts away the details of the actual hardware and makes it possible to have multiple devices in a SmartSpace controlled by a common interface. The agents associated with individual appliances work collaboratively. A

SmartSpace also contains agents that aggregate the control of other agents for purposes of implementing more powerful behaviors.

On a high level, humans and agents interact with each other and trigger events. Human activity is the most obvious trigger for events in a SmartSpaces environment, but the software agents are autonomous and, in addition to being reactive, are capable of taking initiative and performing actions by themselves without human intervention. It's useful to think of software agents as robots without physical form. A properly designed and configured SmartSpace would be able to act on a human's behalf unobtrusively, able to anticipate some of the needs of a home's dwellers before they are even conscious that they need something done.

Currently, Carnegie Mellon Silicon Valley uses JADE (Java Agent DEvelopment Framework) [JADE 2008], as the middleware for SmartSpaces. The physical appliances in the test bed (lights, televisions, DVRs, Roombas, RFID, touchscreen displays, etc.) are all controlled by independent agents which communicate by exchanging messages in the FIPA-compliant agent communication language (ACL) supported by the JADE system. In addition, there are agents for coarser grained entities such as a room agent for a room that contains the appliances and a building agent for a building containing rooms, hallways, etc. The Cricket locationing devices that are the main topic of this paper have their own controlling agent within the SmartSpace environment.

The most important thing that a SmartSpace must know is the location of the humans who dwell within it and the objects in the space. A SmartSpace can make inferences based on location and the context of the current time and previous events. If a person sits in front of a television at a particular time, there is a high likelihood that he or she wants to watch a favorite program. If multiple people sit in front of the TV together, perhaps one person's preferences take priority. (This might be previously determined in a number ways.) If locations of objects and people are available, there are many possibilities for customization to optimize the use of the TV. As another example, we are exploring eldercare health and household chore support. Knowing where an elder, their medication and their groceries are located in the home is important. A possible extension might include telehealth and mobile health applications.

Early research into the problem of indoor location detection was done at Olivetti Research Lab under the banner of the Active Badge project. [Want *et al* 2005] Active Badge used infrared transmitters in an environment studded with receivers. The earliest SmartSpaces experiments involved radio-frequency identification (RFID) tags. Both Active Badge and RFID suffer from the same problem: their resolution is far too coarse to support some of the scenarios SmartSpaces has in mind. At best, they can tell the system if a person or object has entered a fairly wide zone, such as a room or in some cases, in a zone within a room, which is often sufficient for many applications. Basic SmartSpaces scenarios require knowledge of a location in two dimensions. Advanced scenarios could require knowledge of location in three dimensions. This makes it possible to infer, for example, if a person is standing, sitting, or possibly unconscious on the floor.

The Global Positioning System [GPS] provides the type of location information that SmartSpaces requires, but again, the resolution is insufficient for use within a SmartSpace.

Civilian GPS units have a location accuracy of about 10-15 meters, though it is possible to improve on that. Unfortunately, even with improvements, GPS is unsuited for indoor use because of signal attenuation caused by walls and reflections caused by metal components in buildings.

The latest research in an indoor area includes Active Bat [Ward et al 2005], a continuation of the Active Badge project, and Crickets, a hardware and software platform developed at the Massachusetts Institute of Technology. [Balakrishnan et al 2005] [Cricket Project 2005] Both projects used beacons that send out a radio frequency (RF) pulse and an ultrasonic chirp simultaneously. Listener units use the RF to identify individual beacons and use the difference between the arrival times of the RF pulse and the ultrasonic chirp to estimate distances. Knowing the distances between at least three Crickets (and their three dimensional location coordinates) and a broadcasting beacon unit makes it possible for us to calculate the location of the beacon unit within +/- 10-12 cm diameter. The big difference between the Active Bat and Cricket projects is the architecture of the supporting software. Active Bat is highly centralized, while the designers of MIT Crickets chose the opposite approach and emphasized a decentralized system. One reason to prefer decentralization is that it enhances user privacy. Recently, companies such as Cisco and Symbol Technologies (a subsidiary of Motorola) have released commercial systems that promise to track objects and people using a combination of Wi-Fi and RFID. A companion paper [Correa et al 2008] has a description of experiences with Wi-Fi based locationing using signal strength.

We ultimately chose the Cricket ultrasonic location system to provide location information for the current version of the SmartSpace environment. The Cricket hardware is available commercially from Crossbow Technology. A development kit of eight Crickets costs approximately $2000 retail. We also chose not to use the provided demonstration software but to build on top of JADE instead. The net effect was to use Cricket hardware with an Active Bat architecture, that is, to have several cricket listeners (receivers), possible laid out in a grid in the ceiling, concurrently listening to a few cricket beacons (transmitters). The Cricket beacons might be worn, for example, on a person or a mobile appliance (such as the Roomba vacuum cleaner). The distributed listeners each measure the distance to each beacon and report their findings. This aggregates distance measurements and can be used to determine the beacon (and thus the person or object to which it is attached) location.

In the SmartSpaces test bed, three Cricket listeners are mounted on the ceiling of the room and connected via serial ports to a computer running cricket JADE agents. The Cricket agent polls the Cricket ports and extracts beacon distance measurements from the listeners. Three dimensional beacon coordinates are calculated using trilateration [Trilateration], a method of position-finding by measurement of distances (the better-known principle of triangulation refers to position-finding by measurement of angles). The essence of this technique rests in knowing the exact location in 3-space of each listener and the distance (measured via ultrasonics) from the beacon to each of these known listeners. (The sister technique of triangulation uses angle measurements instead of distances.) Trilateration is similar to the method the GPS system uses to determine location, and similar to the method used by cell phone providers to provide emergency locationing information. The Cricket polling agent relays coordinates to interested agents via the JADE framework.

Procedure

Bringing the Cricket-based system up to a usable state was accomplished in four distinct stages:

- *Hardware interfacing*. We attached a cricket to a computer and did initial testing of the hardware to understand the necessary low-level software control.
- *Agent interfacing (Cricket polling agent).* We created a listener Cricket polling agent that contained an implementation of the trilateration algorithm applied to sets of three cricket listening devices and had the agent forward the resulting location information to other agents. At first, we observed a high degree of error in our 3-D measurements. Upon further evaluation and experimentation, we discovered a fundamental problem with the measurements of individual cricket listeners. Each cricket was producing inaccurate data. To compensate for this, we employed linear calibration. All raw Cricket data now passes through this calibration filter. (See "Accuracy Issues" under "Experiences" below.)
- *Control of an iRobot Roomba vacuuming robot with coordinate information*. [iRobot 2005] A Roomba can be controlled externally by sending commands (and receiving polled sensor data) via a bidirectional serial line similar to a Cricket serial line. A Cricket beacon (which does not need a serial connection) is physically attached to a Roomba. In this approach, Roomba motion commands, whose purpose is to reach a particular goal position, are sent to the Roomba via its serial port. These commands are dynamically generated in response to positioning information feedback received from the Cricket agent while still attempting to move to the Roomba's position goal.

A reexamination of cricket accuracy. We did an analysis to determine the cause of location accuracy shortcomings we found with the cricket system. The key problems are the nature of ultrasound in the real world and the directionality of the ultrasound pulses. These include noisy data intrinsic in ultrasonic-based measurement systems. Priyantha describes possible noise sources:

> We have observed the following common ultrasonic sources in an office environment: certain faulty fluorescent lamps, jangling keys, air conditioners, and air conditioning ducts. Since the ultrasonic receiver is a physical resonator that resonates at 40 kHz, any loud noise such a banging door can cause the receiver to resonate at 40 kHz, which produces ultrasonic noise. [Priyantha 2005]

Experiences

1. Serial Interfacing

The first crude experiments involved connecting a single listener Cricket to a USB port using a USB-to-serial adapter. We followed the manufacturer's instructions and tested out the Crickets by commanding the listener with a terminal emulator. We verified that a listener Cricket could 31detect beacon Crickets and return distance information.

2. Java/USB

The next step was to write an agent to interface a constellation of listener Crickets with the JADE framework. JADE is written in Java so we looked for a way to access the USB ports with Java. Unfortunately, more than a decade after the language's release, support for USB ports is primitive. Neither the jUSB nor JSR-80 libraries worked well for USB. Instead, SmartSpaces uses the free-software RxTx library to perform serial communication under Windows XP. Migration to Linux-based systems has been successfully completed. The RxTx library works under Linux as well.

3. Accuracy Issues

Once the initial version of the listener Cricket polling agent was written, we discovered that it was necessary to calibrate the Cricket devices. The distances returned when polling the listeners were consistently short with accuracy decreasing proportionately as distance increased. At one meter the Crickets reported 95 cm. At five meters the Crickets reported 455 cm. In other words, reported distances were short from 5 to 9 percent with the error increasing proportionately with distance. Priyantha conjectures [Priyantha 2005] this increase is due to the ultrasonic signal strength drop off with increasing distance. This results in the detection circuits taking longer time to detect the signal and compute the distance resulting in an increasing positive error.

This matter was addressed by taking careful measurements using an aligned single beacon and single listener. The recorded measurements of the direct distance from 50 cm to 7.5 meters in 10 cm increments were used to build a correction factors table used in the calibration filter. The Cricket polling agent was refactored to calibrate and correct the data via interpolation against the table. Apparently, Priyantha's conjecture might be valid, and fortunately the error is positive linear.

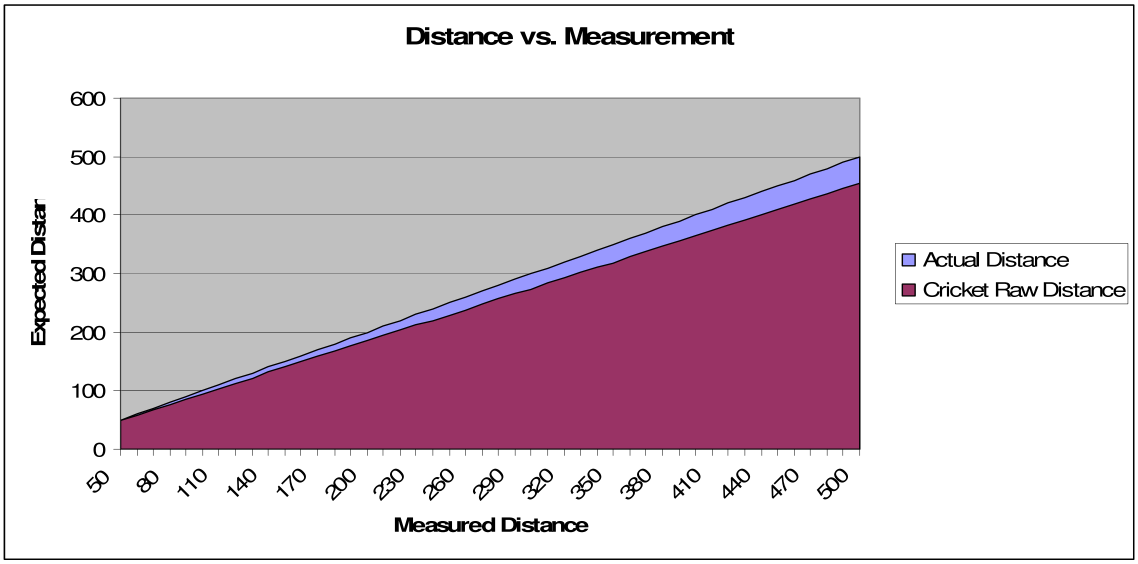

Figure 1: Actual Distance vs. Cricket Measured Distance (in centimeters)

4. 3D Coordinate Generation

Once corrected distances were available, we added a trilateration method to the Cricket polling agent to convert the three distance measurements received into a 3-space coordinate triple, (x, y, z). The mathematics is well known and requires only simple algebra. (See [Trilateration].) The SmartSpaces test bed set up is configured with three listeners at carefully measured locations as reference points. This is the minimal configuration required to make trilateration work. Adding a few additional reference points can increase location reporting accuracy.

The Cricket polling agent reports the calculated coordinates to other agents by sending ACL messaging within the JADE agent framework. In particular, the SmartSpaces Cricket-polling agent sends coordinates to a room agent, which in turn handles the general behavior of the test bed within the room. This in turn forwards location information onto the device agents within the room that require it. The SmartRoomba vacuum cleaner agent is an example one such agent.

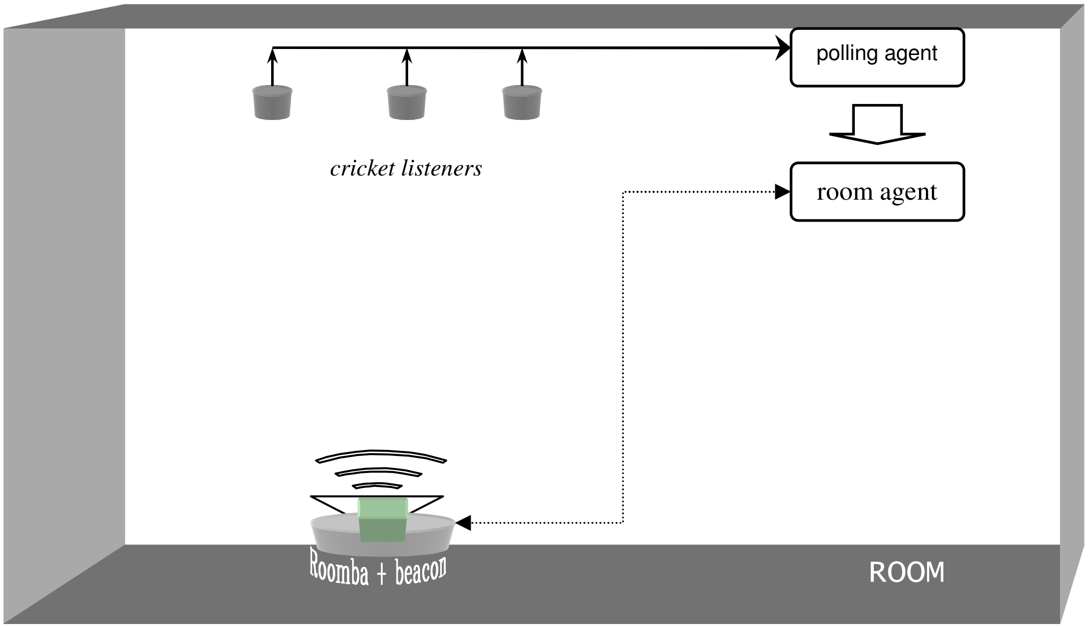

Figure 2. Test setup showing Roomba with beacon, listeners, polling agent and room agent.

5. Platform Testing

As a demonstration platform for proof of 3-D locationing concept, we physically attached a Cricket beacon (without serial line) to the top of a Roomba vacuum cleaner. We commenced a series of very basic tests by placing the vacuum cleaner at the origin of the SmartSpace coordinate system, and faced it towards the test space's positive x-axis. The Roomba's agent was commanded to calibrate the vacuum cleaner's location. In this simple test the Roomba travels along the x-axis to a specified point, then turns completely around and returns back to the origin. The intent of this calibration procedure was to establish possible accuracy of Roomba commanded motion commands. The Roomba performance was inconsistent in this simple test. There were two factors at work: friction and inaccurately detected coordinates.

It was difficult to get the Roomba to perform precise movements to demonstrate locationing. The API for controlling the Roomba depends on accurate motion commands. Unfortunately, the Roomba hardware interprets these movement commands assuming a particular flooring friction coefficient on the turning drive wheels of a known radius. However, the friction for various floor types varies and degrades motion precision. When given a command to move forward a specified distance, the Roomba moves farther on a smooth surface than on carpet. In addition, different varieties of carpet have different coefficients of friction. Friction also has an effect on turns. Telling the Roomba to turn 90 degrees clockwise, then counter-clockwise by the same amount never had the desired effect of returning the Roomba to the original orientation. One final friction related problem is that Roombas have a tendency to veer right when commanded to go straight. This may have something to do with the brushes on the machine.

The Cricket polling agent compensated for most of the friction problem by multiplying the distances and angles sent to the Roomba API with empirically derived constants for the particular surface. Naturally, these correction factors are valid only for the carpet in our particular test bed.

Despite the corrections for friction, the Roomba continued to deviate from the desired performance. Our response was to examine the test bed and the known problems with ultrasonic sensors. Crickets, like most ultrasonic sensors, are sensitive to "noisy" environments (acknowledged by the original developers):

- *Echoes from hard smooth surfaces.* Echoes in the SmartSpaces test bed seemed to be a non-problem. The pulse that Crickets produce is not omnidirectional. Instead, the pulse spreads out in a cone. Since the beacons were on the carpeted floor, this reduced the amount of reflective surface available to pollute the results. Another mitigating factor was that echoes naturally take a longer path and time than a pulse that travels directly. It's easy for the Cricket hardware and supplied firmware to just ignore the echoed pulse as the direct pulse arrives first and this is what is recorded.

- *Limited cone of signal detection.* The signal cone has some interesting implications for the layout of the Cricket listeners. It's possible for a beacon to be either too close or too far from the network of Cricket listeners. If a beacon is too close to one listener Cricket, not all the listener Crickets will be able to detect it. The ultrasound pulse will be outside the cone of detection. See Figure 3a and 3b below to see an example of this phenomenon. If a beacon is too far away from the constellation of listener Crickets, calculating an accurate location becomes problematic because, relatively speaking, all the reported distances will be very similar to each other. Accuracy decreases as the amount of parallax decreases. This can be mitigated to some extent by having a very well defined, regular layout for the listeners.

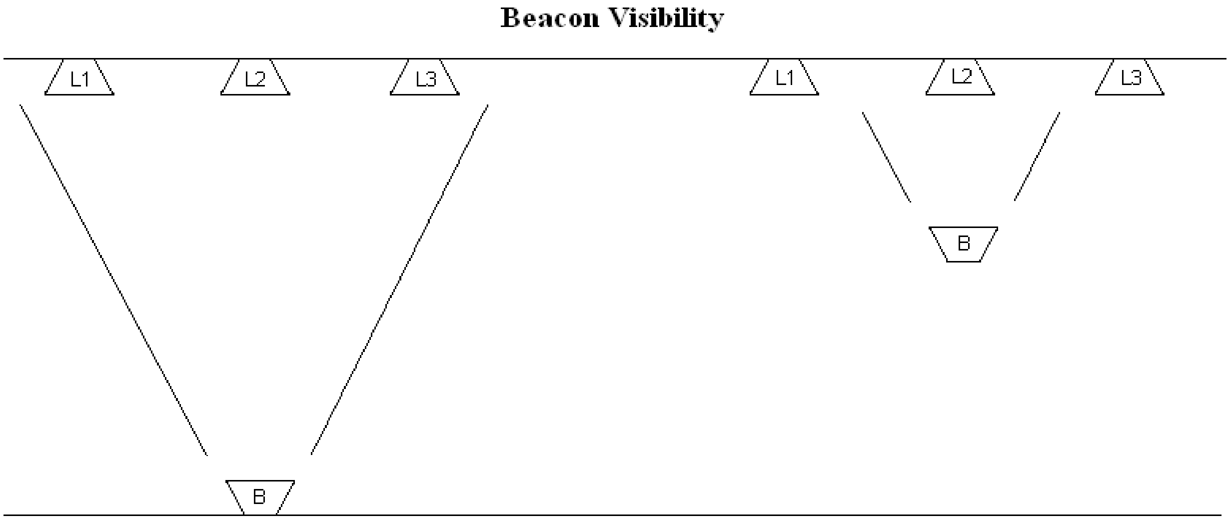

Figure 3a: parallax phenomenon **Figure 3b: listeners too close to beacon**

- *Different distance estimates depending on the location of a beacon within a listener's cone of detection.* The signal cone studies will also help us understand another problem with the Crickets: the varying sensitivity of the listener Crickets depending on the position of any beacon Crickets inside the cone of detection. A beacon in the middle of the cone causes no problem. The listeners seem to have trouble with beacons on the periphery of the cone. Those Crickets report distances that are slightly larger (approximately 3-5%) than they should be. It's possible to optimize the accuracy of a distance reported by any one pair of listener and beacon, but doing so decreases the accuracy of the distances reported by the remaining two listeners. We discovered a discussion of the issue in the definitive MIT PhD thesis on Crickets: Priyantha [Priyantha 2005] describes a similar situation to the increasing distance error (see Accuracy Issues above). That is, as a beacon and receiver are moved off-axis starting from directly pointing at each other so that the beacon is moved towards the periphery of the listener's cone of detection, two things come into play. First, the actual distance between the two increases. And second, the ultrasonic sensor's sensitivity to signals received from increasing peripheral angles rapidly drops of non-linearly. Unless the off-axis angle is somehow determined, this situation probably can not be compensated. With some additional information that might determine this angle from another source, this non-linear error might be compensated in a few iterations.

- *Temperature affecting the speed of sound.* Room temperature has an effect on accuracy. We found that temperature alone could distort measured distances by about 1 cm by affecting the speed of sound. Fortunately, this is a small enough deviation that it is safe to ignore it. Crickets report distances in units of whole centimeters so having results vary by a single centimeter over a distance of a meter (1%) is within experimental measurement error. In addition, the Crickets have temperature compensating built into the chip.

- *Dead zones caused by people or objects obstructing the Crickets*. A larger problem is that Crickets are prone to shadowing and are easily blocked by people or other objects. If two people come together for a conversation, the mere presence of another person might block a clear view of the listener network. Similarly if an object equipped with a cricket goes below a table, it essentially disappears from sight. This doesn't mean that Crickets are useless. One could easily make the same observation about cameras since they depend on visible light.

- *Ambient noise possibly from fluorescent lights or other sources in the environment.* Another problem with Crickets is interference from ambient noise in the environment. We were careful to turn off the fluorescent lights during test runs but despite all precautions, there was a persistent problem with some sort of interference with the ultrasonic pulses. The result was that roughly one percent of the distance measurements were intermittently incorrect, leading to significantly inaccurate reported locations. The cause of the interference is unclear. It is probably something in the environment but we can't rule out a hardware or software fault. This problem should be mitigated by multiple sampling of each point, which currently isn't used.

  Inaccurate coordinates one percent of the time was a problem for Roomba motion because it was sufficient to disturb the machine's determination of its position and its orientation inside the test bed. Once the orientation or location coordinates have been miscalculated, the Roomba could not recover. There are at least four ways to minimize the impact of bad coordinates. First, one could find some way to reject the outlier locations. Second, one could average out the coordinates and take a mean value as the actual location. Unfortunately, it's not clear how to make this work if SmartObjects are constantly moving. Third, we could determine orientation by using two cricket beacons, rather than just one attached to the Roomba. Fourth, a standard approach to this situation found elsewhere is by oversampling and statistically filtering the location data. Location and orientation correction is an area that requires further study.

- *Unstable coordinates*. The final problem we had with the Cricket system was that locations were not stable. That is to say, even if the beacon Crickets were motionless on the floor, there was a minor amount of jitter in the resulting coordinates. Consider the following locations for a motionless beacon that were reported in one experiment:

(-5, -8, 320)
(-3, -8, 320)
(-3, -6, 320)
(-3, -8, 320)
(-5, -8, 320)
(-5, -6, 320)
(-5, -8, 320)

The jitter, in and of itself, is not fatal. We started out each Roomba experiment with a 100 cm calibration step to determine position and orientation before attempting to maneuver to the waypoint. Unfortunately we then instructed the Roomba to make ten centimeter steps toward the waypoint, and then back to the origin. *This* was a problem. Roomba precision was already a problem. The jitter, combined with the tiny steps, made for a setup where accuracy was lost quickly. The correct course of action would have been to make the Roomba take larger steps.

Recommendations

Here is a list of open issues that should be investigated, in order of priority.

1. Determine just how much accuracy is required to achieve the goals of the various SmartSpaces scenarios. Knowing merely that something is in a particular room is enough for some applications, but it is not sufficient for controlling an appliance like a Roomba. Current thinking is that knowing location within a +/-15 cm zone will be sufficient.
2. Isolate the cause of occasional (very) bad distances to be reported.
3. Determine if there is some efficient way to reject bad data in real time.
4. Determine how to calculate the optimal pattern for Cricket listener placement for a particular room.
5. Verify that increasing the step size improves Roomba performance.
6. See if we can achieve continuous tracking of objects by increasing the number of Cricket beacon pings per second. How far can we push the number of pings before running into limits set by the laws of physics?
7. Investigate sensor fusion either by adding redundant Cricket listeners or by adding locationing sensors using a different technology to compensate for some of the distance and angle sources of errors described earlier. Every kind of sensor has some weakness. Using a different kind of sensor to make the same measurement as the original sensor and then fusing the combined data in a complementary way may be able to improve performance. For example using an ultrasonic distance measuring sensor in conjunction with an infra-red or laser distance measuring version could produce better results. If the material of an object does not adequately reflect the signal of one to be measured, it might reflect the other. A glass door probably won't reflect the light, but will reflect the ultrasonic signals. Thus fusing both sets of sensor data, the glass door can be "seen."

8. Determine an easier way to configure and calibrate the test bed. The current setup requires measuring the placement of listeners/reference points by hand with a tape measure. MIT used auto configuration, but it is hard to see how that worked if Crickets are not accurate right out of the box.
9. As with any application of wireless technology, security and privacy issues are important. This experimental test bed only explores the technical aspects in order to explore and understand the underlying and fundamental details. Security and privacy are to be thoroughly considered for this situation before actual deployment is to happen.
10. The experimental test setup used for this investigation had a moving beacon atop the Roomba with fixed listeners at known locations in 3-D space. An alternative configuration (which is often used in a number of the reports and papers found in the cricket literature) is to have several fixed beacons at known 3-D locations and a single moving listener. To achieve this alternate setup in our test bed, a small computer would need to be interfaced to the moving Cricket listener. Both would ride atop the Roomba with the onboard computer performing the trilateration calculations.

Conclusion

Crickets are a comparatively inexpensive way to determine locations inside a building, but they have some inherent problems. That said, there is currently no known sensor that is capable of giving highly accurate readings indoors under normal conditions of anticipated use. Crickets were never envisioned as the ultimate solution for location determination in a SmartSpaces environment. It's better to consider them as part of a suite of solutions that can combine to achieve results that are better than the individual solutions. Until such a time that such sensor fusion based locationing systems is a reality, we can use Crickets for the initial studies of the challenges for implementing the SmartSpaces vision within a normal household setting. Sensor fusion would definitely be desirable. For example, it could be used to augment the cricket locationing with signal strength Wi-Fi locationing [Correa et al 2008] for instances when Roomba movements take it beyond the sensor range of any Crickets in a particular locality.

Acknowledgements

This project was supported in part by Panasonic and Motorola Symbol. The investigation itself benefited greatly from the additional efforts of Martin Griss. In particularly, we wish to recognize Martin's contributions for code development to manage Crickets, to Roomba motion strategy planning and early programming, and for providing extensive advice on the resulting analysis.

References

[Balakrishnan *et al* 2005] "The Cricket Indoor Location System." MIT Computer Science and Artificial Intelligence Laboratory. 2005. http://cricket.csail.mit.edu/.

[Correa et al. 2008] Correa, Joshua, et al. "Room-Level Wi-Fi Location Tracking**.**" Carnegie Mellon Silicon Valley, CyLab Mobility Research Center. MRC-TR-2008-02, Nov. 2008

[Cricket Project 2005] Cricket Project. "Cricket v2 User Manual." MIT Computer Science and Artificial Intelligence Laboratory. Jan. 2005. http://cricket.csail.mit.edu/v2man.pdf.

[GPS] "Global Positioning System." Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Global_Positioning_System

[iRobot 2005] *iRobot® **Roomba**® Serial Command Interface (**SCI**) Specification*, iRobot Company. http://www.irobot.com/images/consumer/hacker/Roomba_SCI_Spec_Manual.pdf .

[JADE 2008] "JADE – Java Agent DEvelopment Framework is an Open Source platform for peer-to-peer agent based applications." Telecom Italia SpA. 4 Nov. 2008. http://jade.tilab.com/

[Priyantha 2005] Priyantha, Nissanka Bodhi. *The Cricket Indoor Location System*. Ph.D. Dissertation, Computer Science and Engineering Department, MIT. June 2005. http://nms.lcs.mit.edu/papers/bodhi-thesis.pdf

[TRILATERATION] "Trilateration." Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Trilateration

[Want *et al*] Want, Roy et al. "The Active Badge Location System." Olivetti Research Ltd. http://www.cl.cam.ac.uk/research/dtg/publications/public/files/tr.92.1.pdf

[Ward *et al* 2005] Ward, Andy et al. "The Bat Ultrasonic Location System." AT&T Laboratories Cambridge. 7 Jan. 2005. http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/