We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Integration of Data Across Disparate Sensing Systems Over Both Time and Space to Design Smart Environments

Peter Bajcsy and Rob Kooper
National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign (UIUC), 1205 W. Clark, Urbana, IL 61801, USA

1. Introduction

Wireless sensing devices are frequently used in smart spaces, ubiquitous and proactive computing, and situation awareness applications (Satyanarayanan 2001), (Vildjiounaite, Malm et al.), (Weiser), (Ilyas & Mahgoub). One could list a plethora of applications suitable for the use of wireless sensor networks and other sensing instruments, for instance, health care (wellness system for aging), environmental monitoring (pollution of air, water, and soil), atmospheric science (severe weather prediction), structural health monitoring (equipment or material fatigue detection), military surveillance (vehicle movement detection), facility monitoring (security and life-cycle of a facility), wild life monitoring (animal migration), or intelligent vehicle design (obstacle detection) (Dishman), (Gupta & Kumar), (Mainwaring, Polastre et al.), (Wang, Estrin et al.), (Roush, Goho et al.), (East), (Rom'an, Hess et al.), (Abowd), (Dey), (Kidd, Orr et al.). The list of on-going projects that include wireless sensor networks and other sensing instrumentation is also growing every day (see NSF, NIST and DARPA projects such as NSF NEON, LOOKING, SCCOOS, ROADNet, USArray, TeraBridge, ORION, CLEANER, NIST SHIELD or DARPA Active Networks, Connectionless networks, DTT). All projects have in common the fact that they represent multi-instrument and multi-sensor systems that can be characterized as smart outdoor, indoor or embedded spaces. The challenge is to build smart spaces that can intelligently sense environments, gather information, integrate information across disparate sensing systems over time, space and measurement, and finally detect and recognize events of interest to trigger event-driven actions. We have been interested in the hazard awareness application scenarios (Bajcsy, Johnson et al. 2008) (Bajcsy, Kooper et al. 2006) that concern humans due to (a) natural disastrous events, (b) failures of human hazard attention or (c) intentional harmful behaviors of humans. Our focus is on the problems related to building hazard aware spaces (HAS) to alert innocent people, similar to the problem related to swimming pool surveillance systems to prevent human drowning (e.g. Poseidon developed by Vision IQ).

While building a real-time HAS system, one has to address the issues of (1) setting up the system to achieve desired accuracy and (2) operating it to achieve reliable performance with or without human intervention. In order to setup a HAS system, one ought to find ways

how to deploy sensors, synchronize them, localize sensors and other instruments in the environment, and calibrate measurements coming from wireless sensors and instruments to obtain values represented in engineering units (for example, a raw value of temperature has to be converted to degrees of Celsius). These steps for HAS preparation allow us to answer questions about when, where and what hazards occur during the operation of a HAS system. In addition, one has to understand the limitations of smart wireless sensor networks (WSN), such as low-power, broadcast range, available on-board memory and CPU, to optimize the layout of sensor networks in terms of minimal wireless loss, minimal energy consumption and maximal information content received from the network. From the perspective of operating a HAS system, the objective is to perform reliable proactive data acquisition, hazard detection, human alert, hazard confirmation and possible understanding of specific hazard characteristics, and finally hazard containment. These building steps have been reflected in our research and development, and are illustrated in the overall HAS schema in Figure 1.

Fig. 1. An overview of several components of the hazard aware spaces (HAS) prototype. The top components represent the setup of HAS while the bottom components correspond to the operation of HAS.

2. Problem description

Research and development of HAS poses several fundamental challenges in the areas of sensing, remote sensor deployment, wireless data acquisition, wireless communication, data integration, distributed signal and image processing, remote and proactive control of sensing, and hazard detection and pattern recognition. It is the understanding of these issues that leads to an optimal real-time system design of HAS. In this chapter, we primarily elaborate on the research themes related to adaptive remote setup of wireless sensor networks. Nonetheless, as one part of the design, we also investigate (a) sensor

measurement accuracy and optimal choice of acquisition parameters, (b) information selection for communication bandwidth control, (c) hazard understanding from sensor and image data, (d) human-computer interfaces for human alert, and (e) the use of robotics in HAS application domains. The research themes map into a development of technology components illustrated in Figure 1 and include (1) deployment of point sensors using remote robot control, (2) synchronization of sensors and cameras, (3) localization of sensors and objects, (4) calibration of measurements from sensors and spectral cameras, (5) proactive camera control, (6) hazard detection, (7) human alert, (8) hazard confirmation and understanding, and (9) hazard containment. The rest of this book chapter presents hardware and software for building a prototype HAS system, and theoretical and experimental solutions to the aforementioned technology components.

3. Hardware and software description

In the HAS system design presented, we used the MICA hardware that is manufactured by Crossbow Inc. The MICA hardware consists of (a) 4MHz Atmega 128L processor, (b) 128K bytes Flash, 4K bytes SRAM and 4K bytes of EEPROM, (c) 916MHz radio transceiver with a maximum data rate of 40Kbits/sec, (d) AA battery pack attached to the processor, and (e) plug-in sensor boards like the MTS101CA, connected through a 51-pin expansion connector. For more details, see (Hightower & Borriello), (Hollar).

The MICA sensors are deployed using an intelligent wheeled robot P2DX8 made by ActivMedia Robotics, Amherst, NH, and an on-board computer-processing unit for real-time processing. The robot has a ring of eight forward sonar sensors that can be used for obstacle avoidance and a two-wheel drive plus balancing caster for smooth motion. The robot is connected to a laptop that is either directly cable-connected to the local area network or wirelessly connected with other computers. In order to control the robotic deployment of MICA sensors, we used keyboard-, gesture- and voice-driven interfaces. For the gesture-driven remote control of the robot, we used the IS-300 Pro Precision Motion Tracker by InterSense, with the update rate of 500 Hz, weight of 15 oz, and angular resolution of 0.02 Deg. It measures yaw, pitch and roll using a miniature solid-state integrated inertial instrument "InertiaCube" and these temporal signals serve as inputs to our gesture recognition algorithm. For the voice-driven remote control, we used wireless audio sensors by Audio-technica Corp. To obtain video feedback from the robot, we mounted a pair of wireless miniature color cameras by Samsung on the robot's platform. To find the location of deployed MICA sensors, we equipped indoor hazard aware spaces with RFID tags and mounted an Alien Technology RFID Reader on the robot to find its location in the building based on detected RFID tags with known locations.

The hazard aware space was also equipped with a visible spectrum camera (Network Color Camera SNC-RZ30N PTZ Pan/Tilt/Zoom by Sony, and Canon PowerShot SD100 digital camera) and by a thermal infrared (IR) camera, (the Omega model by Indigo Systems Corporation, Goleta, CA). The thermal IR camera is a long-wavelength (7.5-13.5 microns) uncooled microbolometer camera designed for infrared applications. It is controlled via RS232 serial port and the analog NTSC video output is digitized using a Hauppauge WinTV board. For temperature calibration experiments, we used a regular thermometer used by chemists as the temperature gauge. It is measuring temperature directly in engineering units of degrees Celsius and providing temperature readings in the range [-40°C, 150°C] with a reading uncertainty equal to ±1°C. A set of preliminary experiments to discriminate burning

materials was performed with a hyperspectral camera by Opto-Knowledge Systems Inc. This camera is based on a liquid crystal tunable filters (LCTF) technology and operates in two wavelength ranges, such as visible [400nm, 720nm] and near infrared [650nm, 1100nm].

4. Localization of sensors and objects

One of the key aspects of any HAS system is the knowledge of where hazards occur based on sensor locations. The general problem of 3-D information recovery from sensor measurements has been addressed in the past by many researchers in the computer vision, machine vision and signal/image processing communities (Dario, Bergamasco et al.), (Marr), (Priyantha, Balakrishnan et al.), (Wechsler) and in the wireless communication community (Whitehouse), (Whitehouse & Jiang). The motivation for obtaining 3-D information often comes from applications that require object identification, recognition and modeling.

In order to detect hazards, one has to deploy sensors in indoor or outdoor environments. The sensor deployment can be achieved manually by a human or automatically by a robot operating in an autonomous mode. The manual deployment can be accomplished by placing sensors either in-situ or remotely with the help of a robot. Every time a sensor is deployed, the problem of localizing the sensor arises since the sensor readings have to be associated with the location for a hazard to be spatially located.

In general, there are three approaches to the sensor localization problem. First, a person who deploys a sensor records also the location and the unique identification (ID) of a sensor. The ID of a sensor is sent with every measurement to a base station where the ID is converted to a location according to a look-up table prepared by a human. Second, a robot that deploys a sensor uploads its location to the database when the sensor is deployed, the ID of the sensor is then sent with every measurement and the location can be found based on the ID of the sensor. Third, a sensor after being deployed determines its own location by communicating with other sensors or beacons with known locations and transmits the location back with each sensor reading. In all three approaches, the localization information could be defined or found in a relative or absolute coordinate system.

The three approaches above have associated tradeoffs between the reuse of sensors and the cost of accurate localization In order to reduce the cost of the system, reusing sensors is preferred over placing sensors once and not moving them again. However, when sensors are relocated in space over time, the manual localization approaches lead to an increased cost of the labour needed to constantly update look-up tables with the new location of the smart sensors. While the sensor-driven auto-localization saves the cost of manual labour, there is a cost in power/energy spent by the smart sensors to run localization algorithms. For instance, in order to perform acoustic time-of-flight localization, all sensors have to communicate using radio and acoustic signals which are very power-consuming operations. The power expenditure leads to frequent replacements of batteries in smart sensors and hence increased costs of human labour and battery replacements. By analyzing the tradeoffs of multiple sensor localization approaches, we decided to explore two solutions based on the approach described as a robotic deployment of sensors.

Next, we outline the two solutions for localizations; one using stereo and a 'smart' sensor network (the MICA sensors) and the other one using Radio Frequency Identification Tags (RFIDs). The first solution aims at accuracy of localization after the sensors have been deployed by combining the results of acoustic time-of-flight ranging and stereo vision algorithms. The second solution aims at power efficiency of localization by deploying RFIDs

in indoor environments prior to hazard sensing and recovering the localization information from the locations of RFIDs. The uniqueness of the RFID-based solution lies in the fact that the tags have fixed locations in the environment, and the reader is mounted on a robot and moving in the environment. This is in contrary to the majority of existing RFID-based applications where the RFID reader is static and the RFID tags are dynamic. After identifying dynamic RFID reader location, the MICA sensors for hazard sensing are uploaded with the location information prior to being deployed, and hence the 'smart' sensor batteries can last longer. Before describing these two solutions as the result of our design optimization analyses, we start by introducing the problems and solutions for remotely controlled sensor deployment using a robot.

4.1 Remotely controlled sensor deployment

The remote robot control system consists of three basic software components including (a) acquisition and recognition of control commands from multiple inputs, (b) client-server network communication, and (c) command fusion and execution by a robot and its arm. We used multiple input modalities to generate the control commands for the robot, such as voice and gesture recognition based controls of a robot. For the voice recognition, we considered wired or wireless microphones. In the gesture recognition system, we used wired orientation sensors mounted on human arms. Finally, our prototype remote robot control system included the mouse and keyboard interfaces, as well as a command line interface with files containing scripted command sequences.

The set of gesture commands is based on the US Navy lexicon for navigating aircrafts on the ground (Lementec & Bajcsy), (Urban, Bajcsy et al.). The set of voice and keyboard commands is user defined and maps in our case to the same US Navy lexicon for navigating aircrafts. Fusion of multiple commands is performed by (a) analyzing time delays and (b) assigning different priorities to commands and the clients issuing those commands (Urban & Bajcsy). Consistent and conflicting commands are considered before a selected command is executed by a robot. For an emergency control, a video signal is sent to a monitoring station. The overview of the system with multiple inputs is presented in Figure 2.
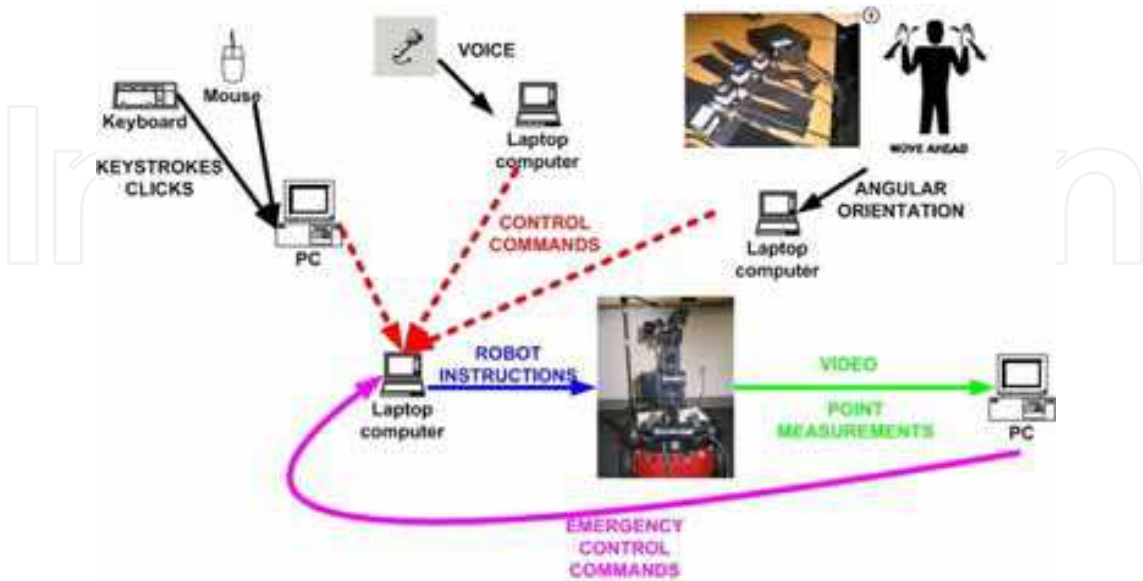

Fig. 2. An overview of a system for remote robot control using sound, gesture and human-computer interface inputs.

First, we implemented an application for the user to control the robot using Human Computer Interfaces (HCI) such as a mouse and a keyboard. A user can use a keyboard and type in commands of his choice and their corresponding parameters.

Second, we developed a template based speech recognition system so that typing can be replaced by more user friendly interface. A database of sound templates is formed by recording commands and extracting Linear Frequency Cepstral Coefficients (LFCCs features) defined by the equation 1, where $i = 0,1,\ldots, P$-1, Y_k is the audio sample at time k, P is the number of LFCCs features equal to 10, and the set of LFCCs is generated for each K= 256 audio sample points. The audio signal is filtered first using a 4th order high pass Chebyshev filter, to reduce the low frequency background noise. Then, the amplitudes of short length, high amplitude blips, glitches, and spikes are reduced to zero before LFCCs features are extracted. We used the Dynamic Time Warping (DTW) algorithm to match any new audio command to all templates previously created. The template with the shortest DTW error distance is selected unless it is above a pre-defined minimum recognition threshold, in which case the input command is classified as unrecognized.

$$LFCC_i = \sum_{K=0}^{K-1} Y_k \cos(\frac{\pi i k}{K}) \tag{1}$$

Third, we added a gesture recognition system to accommodate remote control execution in very noisy environments, for instance, an aircraft carrier deck. While there are many approaches to gesture recognition, we chose to research and develop a solution with active sensors IS300 Pro due to our objective to achieve performance robustness and reliability. Given the choice of an active sensor, our approach to the problem of gesture recognition is based on (1) translating arm motion into a temporal sequence of orientation of angles, (2) describing a sequence of orientation angles with its characteristics, (3) building models of gestures in a lexicon using sequence characteristics of orientation angles, and (4) classifying sequences of orientation angles into gesture classes according to the developed gesture models in real time. The basic premise of our approach is an existence of a unique mapping between human gesture represented by arm movements and a temporal sequence of upper arm and forearm orientation angles. Our approach to robust gesture recognition relies on a two-stage classification technique. The first stage characterizes temporal streams of each Euler angle separately. The second stage uses the combination of Euler angle stream characteristics from the first stage to assign gesture labels according to a set of gesture classification models. At any moment each of the 12 streams of angular values from four orientation sensors are labeled as, steady, oscillating or unclassified, and high, medium-high, medium, low-medium and low. These labels are then use to define each gesture model.

Finally, we enabled robot arm control via mouse and keyboard interfaces in order to perform simple loading and unloading operations. For emergency control purposes, we mounted a wireless camera on the platform of a robot to obtain video feedback[1]. In terms of system architecture, the software is designed based on a client-server paradigm as shown in Figure 3. All input devices (microphones, orientation sensors, keyboard and mouse) are attached to multiple computers that represent the clients in the developed system. In our

---

[1] See a video at http://isda.ncsa.uiuc.edu/gallery.html

laboratory experiments the robot is connected to a laptop using the RS232 connection. This laptop acts as a server and accepts TCP client connections over the network. Each client can issue control commands to the robot by sending commands to the server laptop. The server fuses the commands from all clients, and resolves any conflicts that may occur. After command conflicts are resolved the commands are translated to a set of robot instructions. These instructions are sent to the robot via the RS232 connection, and are then executed.
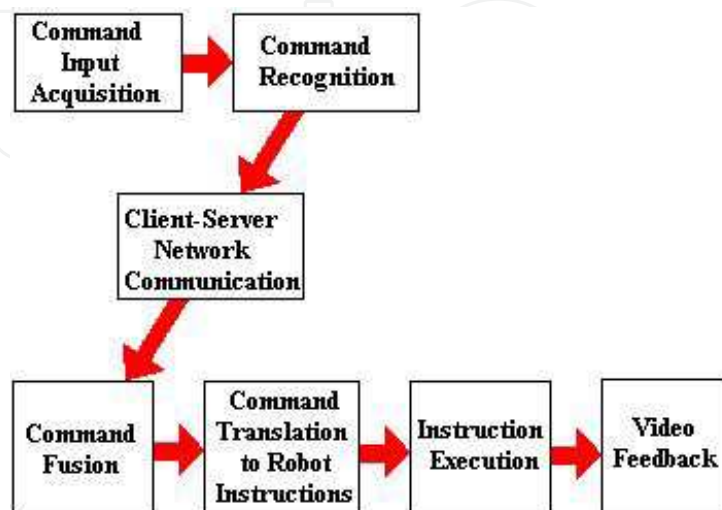
Fig. 3. The signal processing and communication flow of a remote robot control system.

4.2 Location awareness using stereo and MICA sensors
The problem of 3-D information recovery is difficult regardless of whether it addresses static or dynamic object location estimation. In the past, the problem of depth recovery was approached, for example, (a) by vision techniques referred to as shape from cues (Pankanti & Jain) where cues can include stereo, motion, shading, etc., and (b) by communication techniques frequently referred to as location sensing (radio or ultrasound time-of-flight lateration or signal strength analysis (Hightower & Borriello), (Patwari, Ash et al.)). Although the vision and location sensing techniques have been proposed, very few methods are robust and accurate enough to be used in real-time applications. It is well known that many of the depth estimation algorithms are computationally expensive with limited robustness and accuracy in most unconstrained, real-life applications. The need for improved robustness and accuracy of depth estimation motivated our work on stereo and wireless sensor location fusion.

Our approach to the 3D information recovery problem is based on fusing localization data from wireless sensor networks with depth maps obtained through computer vision stereopsis (Scherba & Bajcsy). One could envision performing (1) depth map calibration, (2) sensor location calibration, or (3) depth map and localization fusion. A flowchart depicting the entire process from raw data to calibrated or fused information is shown in Figure 4. We have performed several experiments with synthetic and measured data using the Crossbow MICA2 motes, TinyOS, and Image to Learn (Im2Learn) implementation of the stereo algorithm (Bajcsy, Lee et al.).

Acoustic time-of-flight ranging was implemented according to Figure 5. The first step is to send a message to a ranging endpoint node. The endpoint node, after receiving the message, simultaneously broadcasts a radio ranging message with a 4 kHz chirp. Every node in the
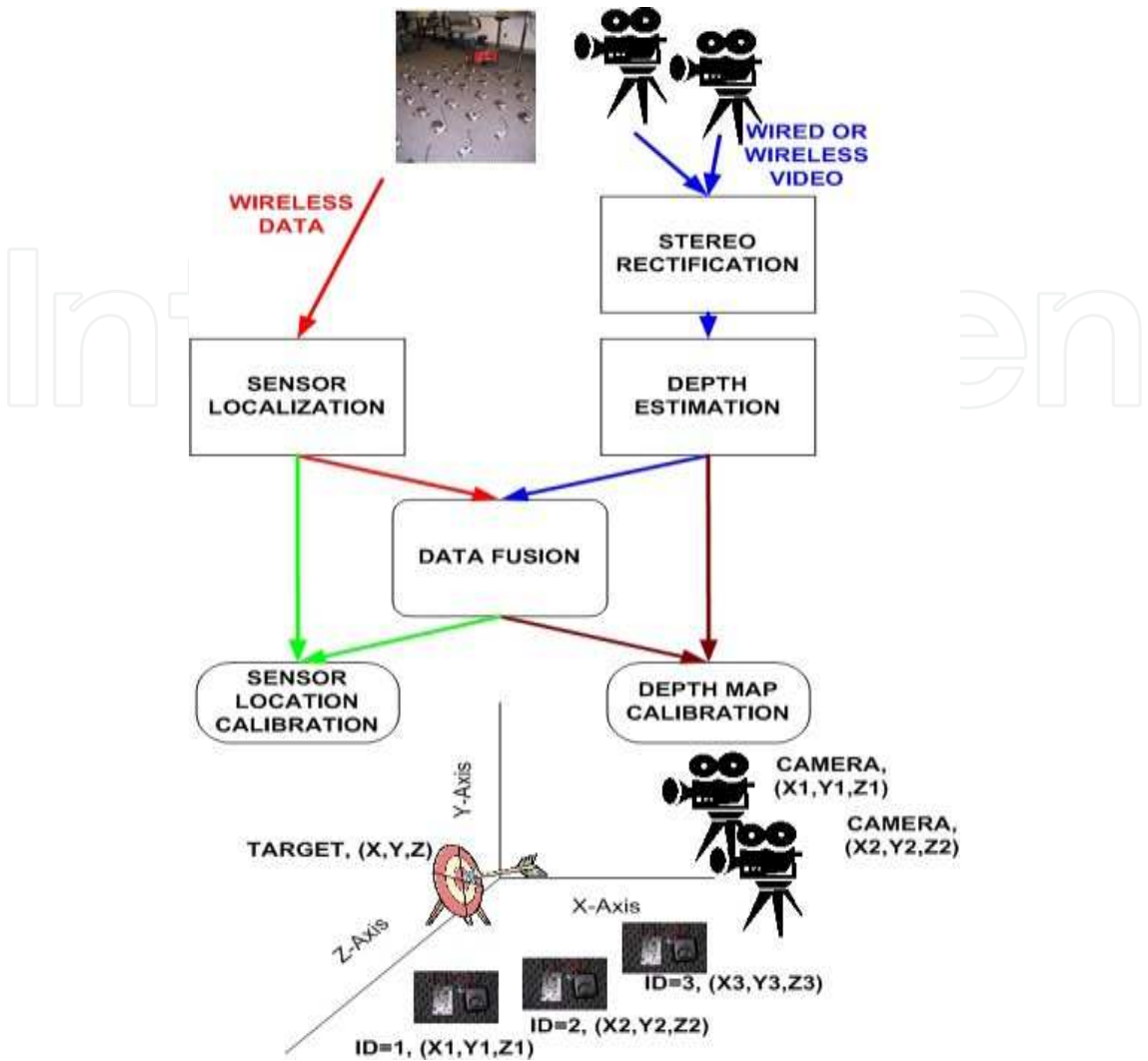
Fig. 4. Flowchart of Sensor Fusion.

network is configured to listen for the radio ranging messages and starts a timer which stops when the audible chirp is heard. A broadcast message announcing the distance between the endpoint and receiving nodes is then sent for all who are interested. Ranging is possible in this setup due to the differential in radio transmission speed (governed by the speed of light, the radio stack, and system-level issues) and the speed of sound in the sensing environment (we use 346.65 m/s for our experiments which corresponds to the speed of sound in air at 25°Celsius). The granularity of the timer on the receiving nodes primarily dictates the uncertainty in the ranging estimates.

Stereopsis is the construction of three-dimensional geometry given multiple views of a scene. The use of stereopsis leads to a depth map that is simply an image of a scene with pixel values given by the depth of each scene point from the camera (the minimum distance from the scene point to the camera plane). We implemented the stereopsis algorithm according to (Hartley). It consists of stereo rectification step (a process which aligns images such that matching points in the resulting images are on the same scanline) followed by pixel matching. In the rectified images, everything can be expressed in terms of disparity that maps inversely to a depth estimate. The stereo algorithm and the acoustic time-of-flight ranging algorithm are described in (Scherba & Bajcsy).
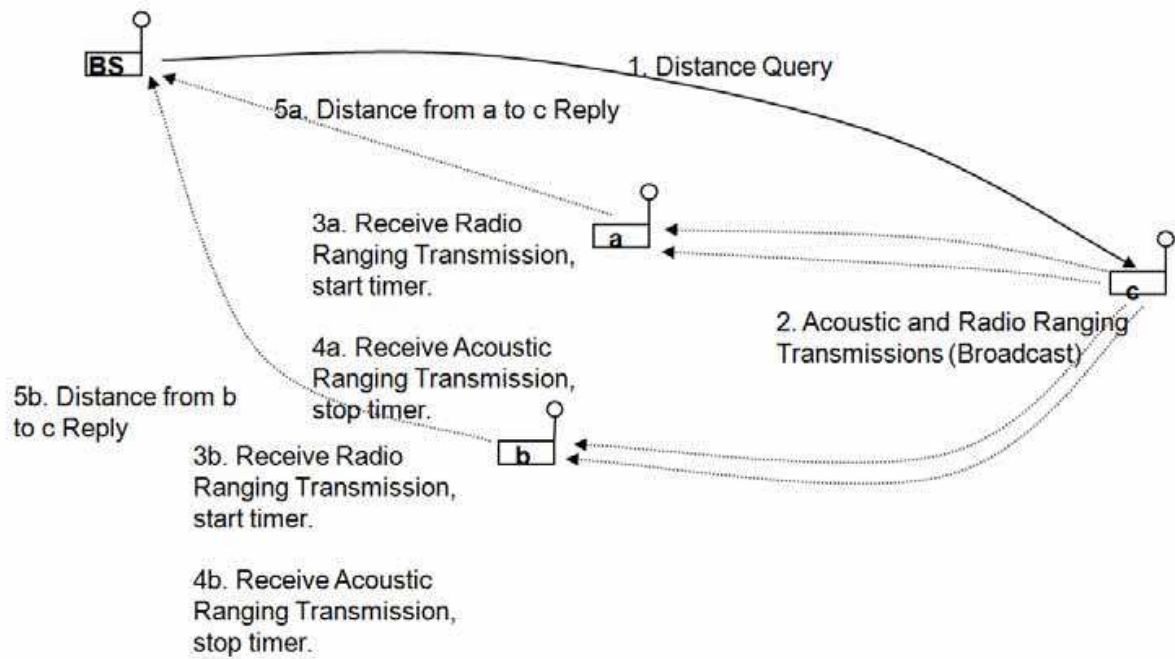
Fig. 5. Acoustic Time-of-Flight Ranging. BS is the base station connected to a computer. The MICA2 motes are labeled a, b and c.

The fusion of sensor localization and stereo depth map results is performed by (a) registering the localization and depth map data, (b) estimating the uncertainty of localization and depth map data as a function depth distance, and (c) fusing the two data by minimizing the uncertainty over the entire depth range. The registration problem is approached by either global optimization or local model-based fitting. The global optimization is achieved by minimizing the difference between depth values and localization values in the least squares sense by solving a non-linear set of equations (number of MICA sensors is equal to the number of equations) using a downhill simplex search. The local model-based fitting approach assumes that a set of *apriori* known sensors is co-planar. Then, the registration is performed by (a) fitting a 3-D surface to a set of *apriori* known co-planar sensor locations, and (b) computing the registration transformation parameters.

In order to fuse the data, the uncertainty of localization and depth map data as a function depth distance is estimated theoretically and verified experimentally. The theoretical uncertainty estimates are derived from a stereo depth disparity equation and from modeling point-point ranging/localization error. Figure 6 shows the determination of a fusion threshold for a particular choice localization and depth map uncertainties. We also developed simulation capabilities for any range of input uncertainties to determine desired fusion thresholds.

To quantify the benefits of fusion, we performed laboratory experiments. The pair of input stereo images is shown in Figure 7 (left) and the MICA sensors were spaced along the depth axis of both cameras. We fused the localization and stereo depth map data sets by minimizing the uncertainty over the entire depth range. The resulting depth map is shown in Figure 7 (right). The accuracy improvements due to fusion for localization and depth map estimation were 95% in our experiments.
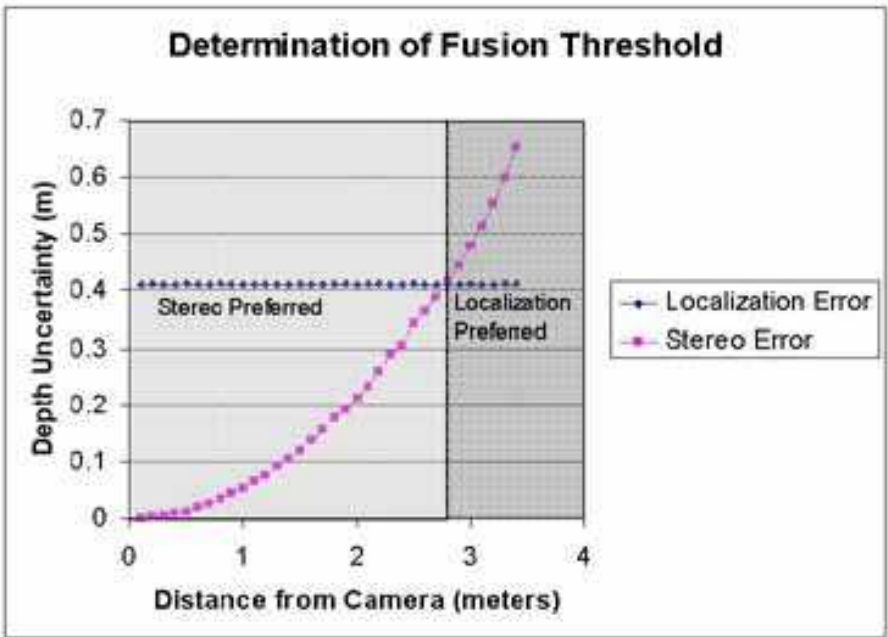
Fig. 6. Fusion decision rule for an image matching error of 4 pixels (stereo error) and a point-to-point ranging error of $\sigma = 0.03m$ (localization error).
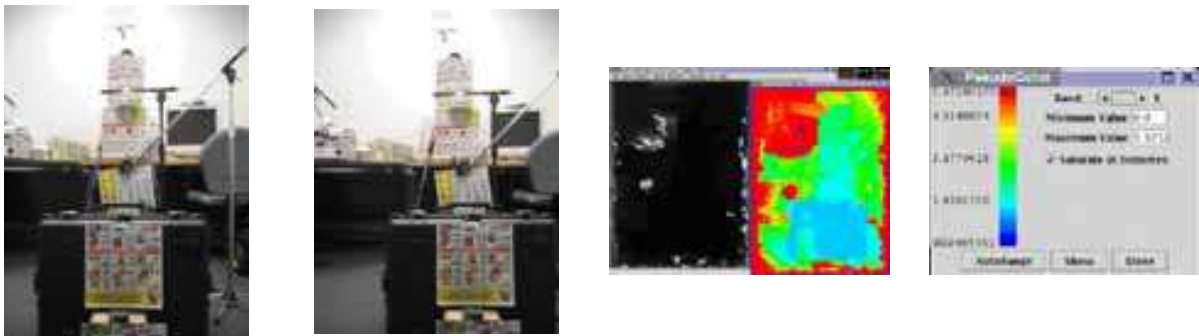

Fig. 7. Left two images show a stereo pair of images taken for quantifying the benefits of fusion in a laboratory setup. To the right from the stereo pair, the images show the resulting depth map after fusing localization and stereo depth map data. Left image is the input depth map before fusion, middle is the pseudo colored depth after fusion, and right is the color legend showing the range of depth values in meters in the pseudo colored depth map.

4.3 Localization with radio frequency identification tags

The aim of this localization approach is to investigate a more power efficient solution by using passive RFID tags compared to the approach using acoustic time-of-flight ranging. Hazard aware spaces might be equipped with passive RFIDs at known fixed locations that could be used for localization of a robot deploying smart MICA sensors. There is a need to explore the cost of deploying RFID tags, their maintenance, robustness and fault-tolerance of reading RFID tags, as well as the accuracy of RFID based localization as a function of the spatial distribution of RFIDs.

In order to evaluate passive RFID technology, we envisioned the following scenarios.

A robot (or person) is moving in a space containing RFID tags, and has a priori knowledge of the passive RFID tag locations. The sensor is an antenna (Alien Reader) that detects tags,

and obtains their ID and possibly other information, like temperature, position, etc. The types of RFID tags we used are shown in Figure 8. Our goal here is to use the RFID Reader to detect tags and, given a previously determined map locations of tags, globally localize itself in the space. The location information is uploaded to a central database and then the MICA sensor is deployed by a robotic arm. Future signals transmitted by the MICA sensors will be tagged with the ID of that sensor and the location can be found in the central database. Additional RFID tags on objects in the room will also be used for robot collision avoidance and for hazard material understanding. Thus, the main focus is to incorporate passive RFID tags into the HAS system by (a) building a sensor model for passive RFID tag detection, and (b) creating a suitable localization algorithm.

Fig. 8. Tag types used in our experimental analysis.

The key experimental elements to be evaluated during testing can be summarized as follows: (a) Reader and tag geometric configuration, (b) tag detection behavior during occlusion, (c) Reader and tag relative speed, (d) tag density, (e) tag type, (f) and Reader type. Our experimental results show that accurate modeling of the RFID Tag and Reader configuration is the key for Tag detection performance for a given type of Reader and Tag, but once this is done, robust global localization can be achieved. Our results also show that occlusion material between the Reader and Tag is not a very large issue unless that material is metallic, in which case tag detection does not occur, or is severely perturbed.

The process of detecting an RFID tag depends on many factors: antenna footprint, the distance between the antenna and tag, and the relative orientation between the antenna and tag. We have identified so far the robustness of RFID tag detection to be dependent on the following main attributes: (1) geometry of RFID tag versus reader configuration, (2) relative motion of RFID tag versus reader and (3) media properties between RFID tag and its reader (e.g., occlusion material). These dependencies introduce the possibility of false-negative and false-positive RFID tag readings. We have ignored in our RFID tag detection model the non-zero likelihood of RFID tag or reader failure and it could be incorporated after a basic probabilistic RFID tag detection model is developed. For now, we assume that the RFID tag type is not a major contributor to the robustness of RFID tag detection.

4.3.1 Related work

A technology that performs a similar function to the localization task that we wish to accomplish is floor-based sensor systems (Kaddoura, King et al.). Using floor-based sensors such as pressure sensors can be quite useful, but they obviously limit use to an environment with an accessible floor. Also, in order to be low-cost, it is likely that floor-based sensing will only be able to provide 2-D tracking, and not orientation tracking, due to the difficulty of attempting to detect in which direction a person or robot is pointing. Other costs of floor-

based sensing involve power usage, wiring, installation of electronics, etc. Also, there are interesting developments in the use of smart sensor technology combined with computer vision for the purpose of tracking objects in a space (El-Zabadani, Helal et al.). The use of passive RFID technology has potential to make simple and robust contributions to such problems, especially when the object being localized is a robot.

Our theoretical framework follows the work of Hahnel et al., (Hahnel, Burgard et al.). The probabilistic sensor model for the RFID antenna was derived for the same technology components (a Pioneer robot and Alien reader/RFID tags) but the geometrical configuration of readers, the number of readers, and the elevation and orientation of RFID tags were different from our experiments (compare Figure 9 and Figure 10). Figure 9 (right) illustrates the angular and range dependency of the RFID tag detection reported in Hahnel et al (Hahnel, Burgard et al.). Also, while Hahnel et al. presented an analysis of the use of augmenting a laser rangefinder with RFID tags during mapping and localization, our objective is to evaluate the strengths and weaknesses of using passive RFID tags as the sole sensor during localization.

Fig. 9. This figure, adopted from (Hahnel, Burgard et al.), illustrates the experimental setup and learning a probabilistic sensor model for the RFID antenna.

4.3.2 RFID tag detection robustness

After preliminary studies of passive RFID technology, we realized the importance of understanding the behavior of RFID tag detection under various conditions, both nominal and off-nominal, for robust performance. Only by examining tag detection behavior will one be able to match the capabilities and limitations of passive RFID technology with their specific application. In our preliminary experiments, we analyzed four main areas: (a) tag detection under varying Tag/Reader geometric configurations, (b) tag detection under varying tag speeds, (c) tag detection during tag occlusion, and (d) tag detection under various tag spatial densities.

Testing tag detection under varying tag/Reader configurations: In this area, we performed a variety of simple experiments, which are summarized below.

- Measurement: lay tag on floor and find average detection range. Result: we found that the average detection range is less than 25 inches.
- Measurement: suspend tag in the air and find average detection range. Result: we found that the average detection range is on the order of 7 feet.

- Measurement: lay tag on floor and find maximum detection range for various angles of the tag with respect to the reader. Result: we found that the maximum detection range varies by approximately 15% depending on the orientation of the tag. This is consistent with the observation that the physical tag is not built with rotational symmetry (i.e. the tag's antenna does have a preferred orientation with respect to the Reader).

Testing tag detection under varying tag speeds: For tag positions in the main detection range of the reader, we found that the Reader detects tag's moving at all practical indoor speeds, with virtually no change in detection rate. We tested this by physically waiving tags, running by the reader, etc. The reader always detected the tag promptly.

Testing tag detection during tag occlusion: We performed simple occlusion experiments which involved completely covering the RFID tag with various materials. We performed these experiments with plastic, wood/paper, and metal. We found that plastic, wood, and paper (up to approximately 2-4 inches thick) have virtually no effect on the nominal detection range, whereas any metal (tinfoil, aluminum sheet metal were specifically tested) covering the tag will completely prevent detection.

Testing Tag Detection with spatially dense tag distribution: Measurement: put roughly 15 tags in the main detection area of the Reader, and see if there is a processing bottle-neck.

Result: We found that when approximately 10 - 15 tags are simultaneously detected, there is a bottle-neck in the Alien Reader processing, and delays of up to 5 seconds occur between reported tag detections (vs. 5-10 ms between reported tag detections when only a few tags are being detected).

To summarize these observations: The dependencies noted above introduce the possibility of both false-negative measurements, in which case a tag is in the expected nominal detection range, but does not get detected, and false-positive RFID tag measurements, in which case the tag is outside the expected nominal detection range, yet still gets detected. In addition to this, it is also noted that depending on the specific tag/Reader configuration, extremely different tag detection behavior can occur. Thus, the tag/Reader configuration is probably the most important factor during the detection process (for a given type of RFID Tag). This complex behavior requires one to spend much effort in building an accurate sensor model, which is discussed in the next section.

4.3.3 Building a sensor model

As will be discussed further later, it is common during probabilistic localization to make use of a sensor model which specifies a likelihood of a sensor measurement given the position of the robot (or person). In this section we present a methodology for developing such a sensor model for use during localization with passive RFID technology. As noted above in discussion regarding tag detection under varying Tag/Reader configurations, it is necessary to build a sensor model which is specifically appropriate for the Tag/Reader geometries and configurations that will occur during deployment of the tracking/localization system. This is because the tag detection behavior changes drastically between different Reader/tag configurations. A simple model will not be sufficient if one's actual implementation differs in any number of fundamental ways, some of which are extremely difficult to foresee before the actual implementation. For example, in our experiments, we mainly worked with RFID Tags distributed in a grid on a carpeted floor, which leads to a very different sensor model than that obtained in (Hahnel, Burgard et al.) (compare Figure 9 and Figure 10). When laid out on a carpeted floor, the maximum read range was roughly 90 cm. In contrast, when tags are held in the air in front of the Reader, the maximum read range is on the order of 200 cm.

This behavior led us to develop the following methodology for building a sensor model. First, one must determine the possible Reader/tag configurations and geometries that will arise in their given implementation. Next, one must fix the Reader's location, and distribute a spatially dense array of tags in the very configurations and geometries that are expected. This array of tags is then used to determine the "footprint" of the Reader, and more specifically, the likelihood of a tag detection given the relative position of the tag and Reader. The sensor model then becomes a manifold of detection likelihoods, which can be used directly in the localization algorithm described below.

In our experimental analysis, we followed this procedure with great success. We distributed tags in front of the Reader (Figure 10 left), and collected statistics on tag detection while varying the angle of the Reader. We then used that information to derive the manifold of detection likelihoods over the space of possible geometries.

Fig. 10. Left: documentation of setup used to build the sensor model. Right: Image showing nominal detection range when Tags are distributed on the floor and the Reader is mounted on the robot and tilted forward 30 degrees. The 4th "row" has a radius of rough

4.3.4 Probabilistic localization

There are many methods available to perform localization and tracking. We present one probabilistic localization method here, in order to quickly evaluate the accuracy and robustness of using RFID technology as the sole sensing capability during localization. The algorithm we used to perform probabilistic localization, derived from (Choset, Hutchinson et al.), is very simple, but also very computationally expensive. The problem of localization is to estimate the state $x \in X$ of the robot, where X is a discretized state space consisting of possible 2-D positions and orientations. So, for each time step k, we estimate the *posterior probability* $P(x(k)|u(0{:}k\text{-}1), y(1{:}k))$ over all possible states, where $y(1{:}k)$ represents the sensor measurements obtained at times $1,\dots, k$, and $u(0{:}k\text{-}1)$ represents motions taken at times $1,\dots, k-1$. Note that during a given time step, k, it is possible to detect multiple RFID tags, where the likelihood of a tag_m detection is independent of whether or not tag_n was detected (for all m, n such that $m \neq n$). Also note that as background knowledge throughout this analysis, we have a given map of RFID tags.

The key to this probabilistic localization scheme is the recursive Bayesian filtering equation, which specifies how to use the last estimate (the *prior*), in conjunction with current sensor

measurements and assumptions about current motions, to calculate a current estimate of the robot's state (the posterior):

$$P(x(k)\,|\,u(0:k-1),y(1:k)) =$$
$$\eta(k)P(y(k)\,|\,x(k))\Sigma_{x(k-1)\in X}(P(x(k)\,|\,u(k-1),x(k-1))P(x(k-1)\,|\,u(0:k-2)y(1:k-1)) \tag{2}$$

Above, $P(x(k-1)\,|\,u(0:k-2)y(1:k-1))$ represents the prior, or the probability that the robot is at location $x(k-1)$ before motions or sensor measurements are taken into consideration. The term $P(y(k)\,|\,x(k))$ is called the *sensor model*, which represents the likelihood of the measurement $y(k)$ given the robot (or person) is at location $x(k)$; and the term $P(x(k)\,|\,u(k-1),x(k-1))$ is called the *motion model*. In our application, we used a Gaussian distribution centered on $x(k-1)$ as the motion model, and the sensor model is determined as according to the procedures described in the previous section. The value $\eta(k)$ is normalization constant to ensure that $P(x(k)\,|\,u(0:k-1),y(1:k))$ sums up to one over all $x(k)$. We adopted an algorithm from (Choset, Hutchinson et al.), pg. 313, which we used to compute the above posterior at each time step. We initialized the prior to be a uniform distribution across all possible states, which allows us to perform a global localization.

Note that, because we are representing possible states using a discrete grid, the above algorithm leads to $O(N^2)$ computational complexity, where N is the number of points in the grid (Choset, Hutchinson et al.). In the near future, we would like to update the algorithm to improve efficiency to allow larger state spaces and finer time discretization. This can be done by using particle filters and Monte Carlo localization techniques (Choset, Hutchinson et al.),(Dellaert, Fox et al.).

4.3.5 Experimental results

We implemented the approach described above using one Alien Technology RFID Reader and tags of the type shown in Figure 8. We arranged a grid of RFID tags as shown in Figure 11, loaded the Pioneer robot™ platform with the RFID Reader, and moved it in a pre-defined path. Note that in this configuration, the maximum detection range is approximately 90 cm. Ground truth measurements were acquired throughout the experiments by keeping track of the times at which the robot passed certain pre-determined positions. Note that these ground truth measurements include relatively large errors of up to 13 cm.

Our short term goal was to detect at least one RFID every 100 ms while moving the robot through the grid of tags. We found that using a tag spacing of approximately 40 cm allowed us to maintain this detection rate. Figure 12 show an example of the experimental data, and the resulting localization performance. Our localization algorithm quickly arrived at the robots position, and was able to track the movement of the robot very well. Figure 13 shows the localization error throughout one experiment. We believe that much of the variability in the error is due to inaccuracies during ground truth observations. Despite this, we were able to quickly achieve and maintain a fairly smooth position estimate over time to within an average of roughly 20 cm of the actual position.

Fig. 11. Experimental setup. The Alien Reader was mounted on the platform of a robot (left). The RFID tags were placed on the floor in a regular grid pattern (right).

Fig. 12. These images show three time steps in the middle of an experimental run. Open circles represent undetected tags. Filled blue circles represent detected tags. The red circle represents the average robot location, and a green circle represents a ground
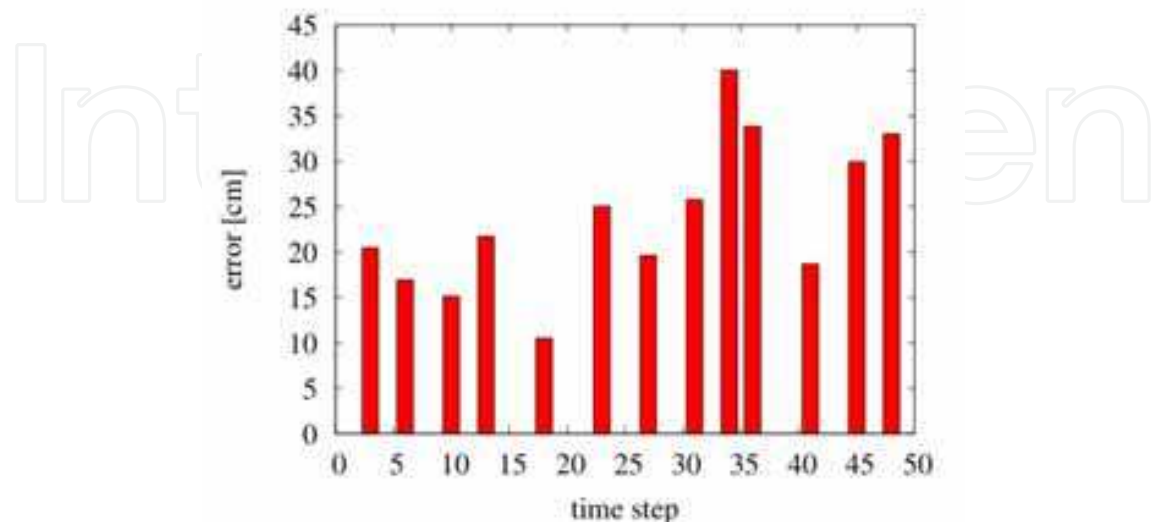

Fig. 13. Plot of 2-D position error over the course of an experiment. Note that our ground truth measurement was extremely noisy, having an accuracy of only +/- 13 cm.

5. Integration of data across disparate sensing systems over time

When integrating data across disparate sensing systems, time will play a crucial point. The readings from different sensing systems need to be time stamped to enable integration. Furthermore, the time stamps need to be with respect to a common clock. It is also important to look at the frequency of updates. When dealing with battery powered devices, like the MICA sensors, we need to consider the tradeoffs between the frequency of sending updates and the power consumption required. In addition, the frequency of sending wireless updates is also correlated with collisions in the sensor network and hence has to be understood.

5.1 Synchronization of sensors and cameras

In order to obtain useful sensor readings, one would like to know when the sensor readings were taken. Thus, our goal was to synchronize cameras with the deployed MICA sensors, or the MICA sensors with cameras. To achieve the synchronization goal we have to collect the data from both cameras and MICA sensors with accurate timestamps. These timestamps combined with the location of each sensor and camera would allow us to associate a MICA sensor value with an image pixel value. Based on our synchronization needs, we designed and developed a temporal calibration technique that follows the schema in Figure 14.

Cameras are attached to a personal computer (PC) and the PC uses its internal clock to set a timestamp for every captured image (the time the first bit of the image is received). If cameras are attached to multiple PCs then the PCs are synchronized using the standard NTP synchronization protocol (Mills). The MICA sensors communicate through an interface board with either the same PC as one of the cameras, or a different PC. In the case of a different PC we assume the camera PCs and the MICA PC are synchronized using the standard NTP synchronization protocol.
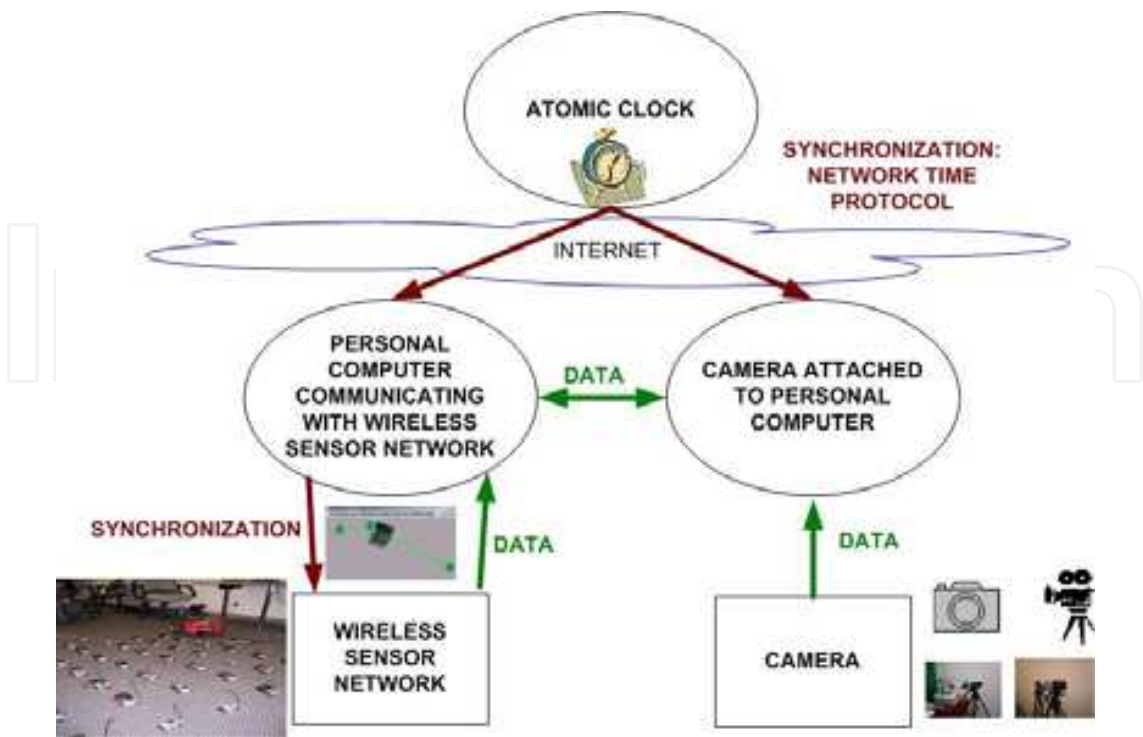

Fig. 14. A schema for time synchronization of MICA sensors and cameras.

The MICA sensors can be viewed as small autonomous PCs. We need to synchronize the time of these small PCs with the PC receiving data. One approach is let the PC receiving data time-stamp the incoming messages. However, if there are some delays in the network, delays due to processing of the incoming messages, or if the MICA sensor buffers multiple readings for their later transmission as a single packet, then the timestamps of the MICA sensor readings would be incorrect. We approached this problem by implementing a simple time synchronization of the MICA sensors with the PC. Since the MICA sensors use a wireless ad-hoc network to send messages back to the base station, we decided to leverage the inherent broadcast of the wireless network to do the time synchronization and not have each of the MICA sensors run the NTP synchronization algorithm.

At the MICA initialization, the PC will send the current time to each of the MICA sensors. A MICA sensor will update this timestamp every 10ms. Each reading on the MICA sensor will be sent back to the PC with the timestamp of the reading. The clocks on the MICA sensors are not as accurate as those of the PC. To prevent too much drift between the MICA sensor and the PC the timestamp on each incoming packet is compared with the current time of the PC, and if the difference between these two is too large (350ms in our case) then the PC will send new time to re-synchronize the MICA sensor.

5.2 Frequency of sensor update

There are several challenges when it comes to (1) acquiring data continuously from wireless sensor networks, (2) dealing with large numbers and high spatial density sensor networks, and (3) performing multi-instrument integration tasks in real time. We investigated experimentally the impact of several communication protocols, spatial sensor arrangements, MICA antenna orientations, presence of other wireless devices, acquisition sampling rates and the dependencies of the number of active MICA sensors on the wireless information loss. Our results were summarized in our past publications (Bajcsy, Kooper et al.), (Saha & Bajcsy), (Scherba and Bajcsy). Next, we only briefly describe the issue related to acquisition sampling rate.

Figure 15 shows how the number of samples received per second decreases with an increasing number of sensors. If there is only one sensor and we sample every 128ms, then we receive almost eight readings per second (1000/128). If we add more sensors then the number of readings decreases because of the collisions in the network. With seven sensors transmitting we obtain the same number of samples per second using 128ms sample rate (and a lot of collisions in the network) or 256ms sample rate. If we would increase the number of sensors then we would continue finding these tradeoff configurations (number of sensors and sample rate configuration assessed by the number of reading lost due to network congestion). From these experimental studies one can make determinations when it is better to switch to a lower sampling rate and still receive the same number of samples per second as at the higher sample rate (while saving battery power). Although unclear from these experiments, it is hypothetically possible that with 128ms sample rate and many sensors in the network one would receive fewer samples per second than with the same number of sensors and 256ms while wasting battery power. Thus, knowing the optimal sampling rate for a given number of active sensors will not only decrease the number of collisions in the network but also save sensor energy. The MICA sensors are battery operated and have a limited supply of energy. Transmission is in most cases the most expensive operation the sensor performs and thus minimizing the number of transmissions will increase the lifetime of the sensors.
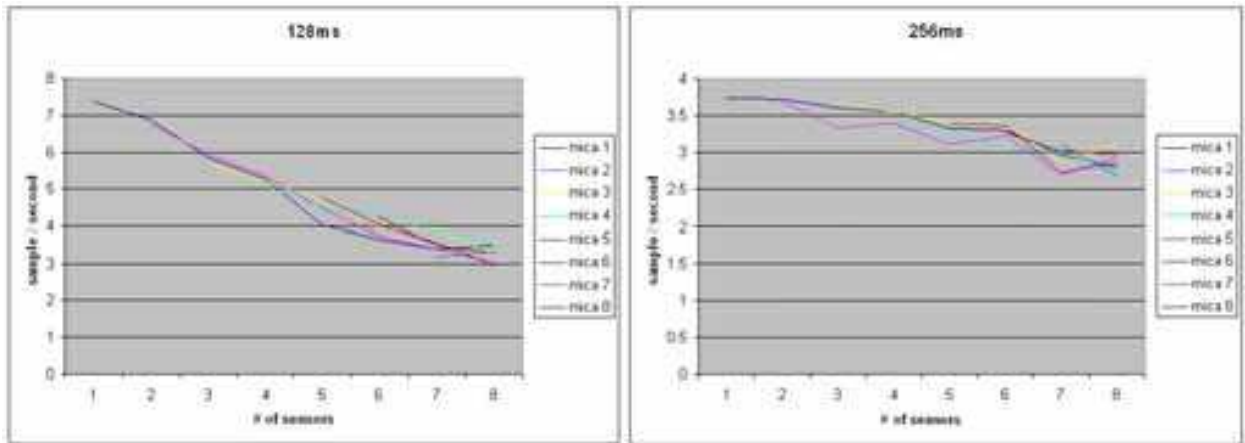
Fig. 15. Samples (sensor readings) received per second as a function of the number of sensors for the temporal sampling rate equal to 128ms (top) and 256 ms (bottom).

6. Calibration of measurements from sensors and spectral cameras

Our objective is to detect and recognize hazards such as fire. We strived to perform continuous wide area monitoring using thermal IR and visible spectrum cameras. Many building already have low resolution visible spectrum cameras to monitor security in the building. Our hazard aware system could use this existing network of cameras to do a low resolution detection of any hazards. Once a hazard has been detected we use the robot to place higher accuracy sensors in the environment to confirm the hazard. In case of a false alarm we want to recalibrate the system to take the new data into account. To be able to achieve this goal we need to integrate the readings from the visible and thermal IR cameras with those from the MICA sensors.

To get accurate hazard detection from raw sensor readings and camera pixel values, for example from the thermal IR camera, one has to convert sensor and camera raw values into engineering units, such as degrees of Celsius or Fahrenheit or Kelvin, otherwise the raw values cannot be used for detection and recognition purposes. This conversion is also denoted as a spectral calibration since temperature (a variable representing thermal wavelength range) could be replaced by any other spectral variable, for instance, a variable that represents visible spectrum, near infrared or radar wavelengths. Figure 16 shows our proposed schema for calibrating raw values.

First, we explored the calibration of thermal IR images using pre-calibrated MICA sensor readings. One can find the need for thermal IR camera calibration in many other areas, for instance, in remote sensing (radiometric and photogrammetry calibration of aerial and satellite imagery), robotics (vegetation detection using near calibrated infrared and red wavelength imagery), astronomy (brightness estimation of stars using thermal IR imaging of the sky) or military (battlefield analyses). We foresee the use of widely distributed and deeply embedded "smart" micro electro-mechanical systems (MEMS) sensors as potential thermal IR calibration gauges for thermal IR cameras in future.

The calibration procedure can be described as follows assuming that all sensors and cameras were synchronized and their locations are known. First, MICA sensors are programmed to sense and send temperature readings over a certain time period. Second, during the same time period, temperature measurements are collected with a thermometer (a calibration gauge). Third, a calibration transformation is established for MICA temperature sensors
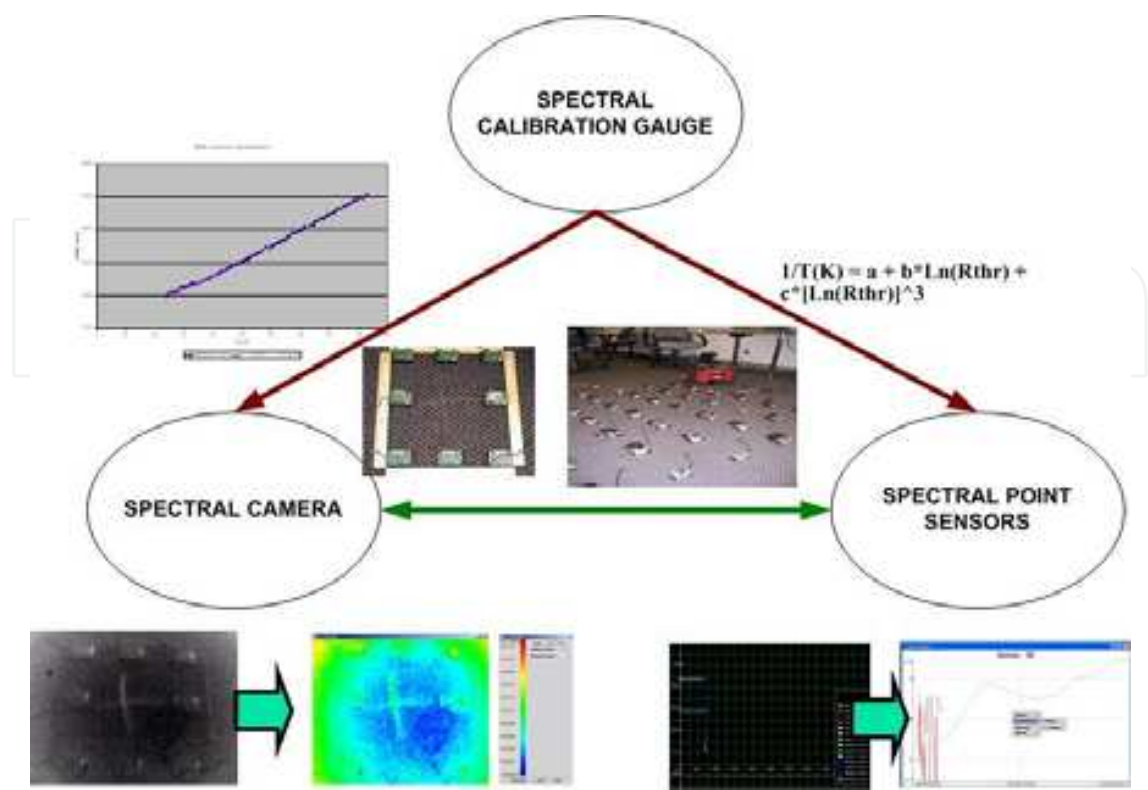
Fig. 16. A calibration schema for MICA sensors and spectral cameras using an appropriate spectral gauge. We used temperature sensors on a MICA sensor board, thermal IR camera and a thermometer as a calibration gauge in our experiments.

using a factory recommended formula and verified with thermometer measurements. Fourth, both thermal IR camera and MICA sensors are initiated to acquire data by broadcasting a RESET signal to MICA sensors and triggering thermal IR camera acquisition. Fifth, MICA sensors transmit every set (packet) of temperature measurements with the state of the internal counter (time stamp) to the base station attached to a personal computer (PC). In meantime, the thermal IR camera acquires data with the time stamp of the CPU clock counting from the RESET signal. Sixth, the MICA raw temperature measurements are received and transformed into degrees Celsius. Seventh, MICA temperature sensor locations in the thermal IR image are identified, and statistics of the transformed MICA temperature measurements and the thermal IR image pixel values at the MICA sensor locations are related to form the final calibration transformation. In this step, if the entire scene viewed by a thermal IR camera is temperature homogeneous then MICA temperature sensor locations in the thermal IR image do not have to be identified and statistics of the thermal IR image can be computed over the entire image.

Second, we explored the calibration of MICA sensor readings using pre-calibrated thermal IR images. This problem turns out to have a great application for large scale sensor deployment scenarios. Calibrating a large number of MICA sensors is a very tedious and time-consuming process since every sensor has to be treated separately. One should be aware that although MICA sensors come with a manufacturer's recommended calibration formula, each sensor has its own hardware characteristics. This is illustrated in Figure 17 for

eight MICA sensors in the close proximity of the base station (the PC receiving data). The raw values vary significantly in terms of their amplitude (vertical axis) and slightly over time (horizontal axis).
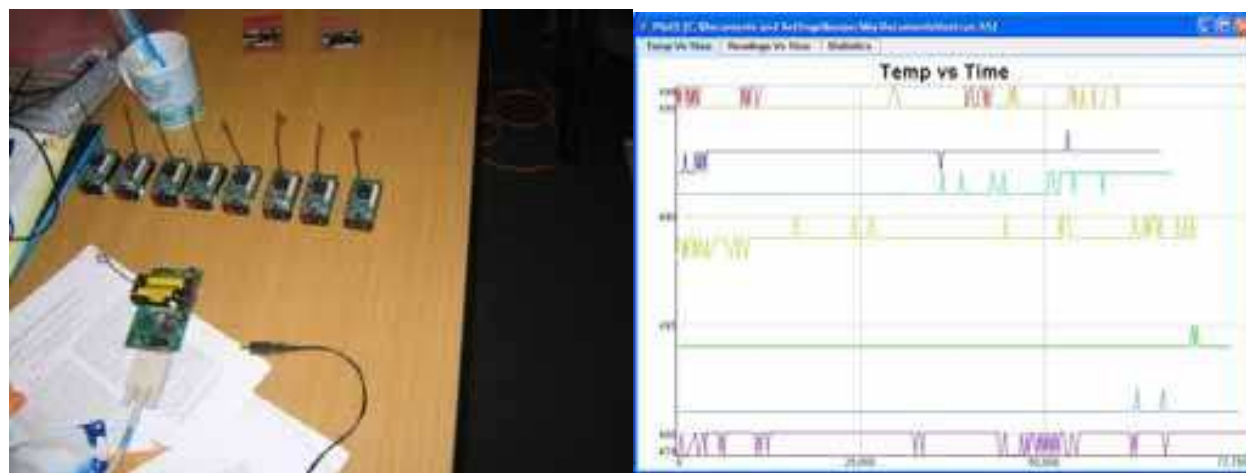

Fig. 17. Left - MICA sensor spatial arrangement. Right - Variations of raw temperature readings obtained from the MICA sensors shown in the left picture.

In this case, the calibration procedure can be described as follows. First, we calibrate a thermal IR camera by acquiring thermal infrared images of a blackbody, such as a cup of hot water, while measuring its temperature with a regular thermometer. Second, both thermal IR camera and MICA sensors are initiated to acquire data by broadcasting a RESET signal to MICA sensors and triggering thermal IR camera acquisition. Third, MICA sensors transmit every set (packet) of temperature measurements with the state of the internal counter (time stamp) to the base station attached to a personal computer (PC). In meantime, the thermal IR camera acquires data with the time stamp of the CPU clock counting from the RESET signal. Fourth, the MICA raw temperature measurements are received and transformed into degrees Celsius. Fifth, MICA temperature sensor locations in the thermal IR image are identified. All calibrated thermal IR image pixel values at the MICA sensor locations are used to form the spatially dependent calibration transformation for MICA sensor readings. The details of the calibration procedures can be found in (Bajcsy, Kooper et al. 2006).

7. Hazard detection, confirmation, understanding and containment

Hazard detection is achieved by comparing calibrated temperatures with a pre-defined hazardous temperature. When a predefined hazardous temperature is exceeded on any sensor, MICA, thermal IR and visible spectrum, the frames are analyzed to detect 2D characteristics (location, shape, color, texture and temporal signatures). We implemented temperature-based thermal IR and color-based visible spectrum image hazard detections from video streams as illustrated in Figure 18. This type of detection guides hazard confirmation step which is performed by deploying more MICA sensors, or by zooming and panning cameras according to the operator's needs.

It is important not only to detect fire hazards, but also to understand what material is burning in order to bring the appropriate extinguishers and to discriminate safe from

dangerous fires, e.g., a couch burning versus wood burning in a fire place. To understand what material is burning, one could analyze hyperspectral images of hazards together with thermal IR images. The assumption of this approach to hazard understanding is that temperature and high dimensional visible spectrum signature of fire flames uniquely defines burning materials. Finally, hazard containment requires studying hazard types, containment methods, hazard accessibility and many other application specific constraints. We have prototyped a demonstration showing that a robot could be used for the purpose of hazard containment when fire hazards in an office building are still relatively small in their extent. It is our belief that constant monitoring of hazards and an immediate hazard containment action could prevent large scale fire hazards and significant financial damages.

Fig. 18. An illustration of color-based visible spectrum image hazard detections from a video stream. Red crosses are placed over candle flames in three frames extracted from a video sequence.

7.1 Proactive camera control and human alert

Once a hazard has been confirmed a human needs to be alerted. We have investigated proactive approaches to camera control, spectral image analysis and human alert mechanisms. We have completed a design phase of a proactive camera control system that can trigger visible spectrum and thermal infrared spectrum cameras based on luminance and temperature sensors mounted on the available MICA sensor boards. Our current design addresses the problems of (a) efficient bandwidth management by proactive camera control (low bandwidth monitoring of hazard awareness spaces with MICA sensors and high bandwidth monitoring of hazard awareness spaces with cameras), (b) hazard understanding (multi-spectral sensing including visible and thermal infrared information) (c) choice of the best spectral modality to capture data about the environment based on the data provided by the MICA sensors and (d) human alert mechanism (image analysis to highlight areas of potential hazard). The proactive camera logic can be described as follows:

• If light is on then visible camera shown.
• If light is off then thermal IR camera shown.
• If temp > thresh & light is on then visible camera & hazard region enhancement are shown.
• If temp > thresh & light is off then thermal IR camera & hazard region enhancement are shown.

Examples of proactive camera control and human hazard alert mechanism are shown in Figure 19. Note in Figure 19 how the system uses the RGB camera when the office light is

on, and combines the information about white (hot) regions from the thermal IR camera to highlight the RGB region of interest (the hazard region).
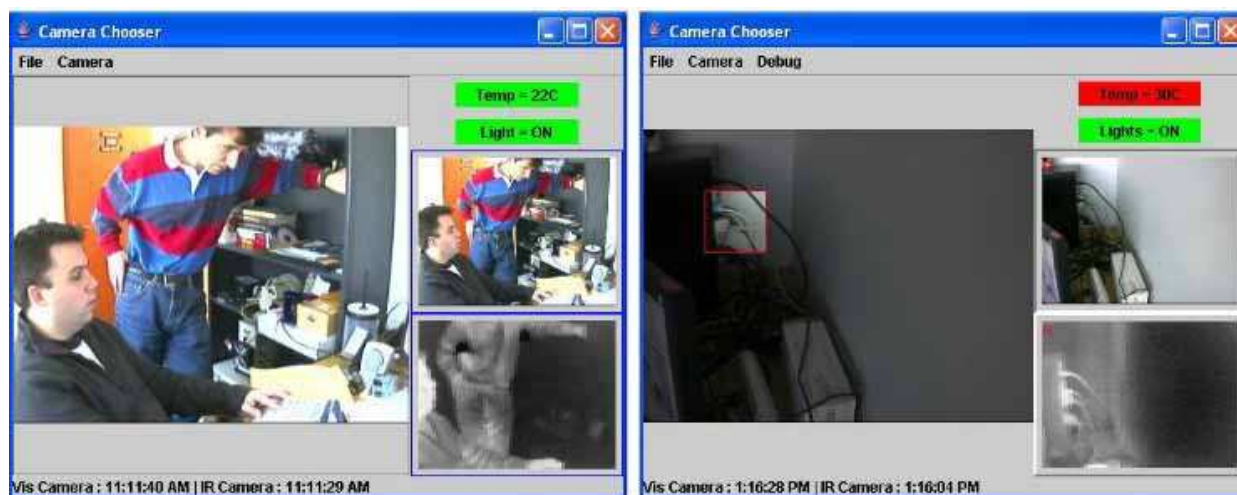

Fig. 19. Left - Proactive camera control. Depending on the temperature and light readings from a wireless network of motes, either a visible spectrum video or a thermal infrared spectrum video is sent to a hazard monitoring station. Right - Illustration of hazard alert by automated region selection and image enhancement.

8. Summary

We presented several research problems and their solutions developed for building hazard aware spaces. We have developed a method for deploying sensors remotely with a robot. To allow us to remotely control the robot we have looked at different modalities of controlling the robot (e.g. voice, gestures, a mouse or a keyboard) and addressed the problems related to fusing multiple simultaneous commands. In multi-sensor and multi-instrument systems like the HAS systems, we have investigated methods how to synchronize, localize and calibrate data. After analyzing the tradeoffs of localization approaches, we explored the effectiveness of using passive RFID technology for localization. We analyzed what elements most effected RFID tag detection robustness, and presented a methodology for building a sensor model for accurate RFID-based localization in addition to the standard acoustic time-of-flight ranging and stereo vision localization methods.

Once the sensors have been configured, we fused the incoming data and maximized our information gain from disparate sensing capabilities. Finally, we showed proactive approaches to camera control, spectral image analysis and human alert mechanisms, and briefly outlined a few challenges in application scenarios similar to the hazard aware space.

9. Acknowledgements

10. References

Abowd, G. D. (1999). "Classroom 2000: An experiment with the instrumentation of a living educational environment." IBM Systems Journal 38 1999: 508-530.

Bajcsy, P., M. Johnson, et al. (2008). Real time localization and 3D depth estimation across disparate sensing systems: Toward hazard aware and tele-immersive spaces. Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, Bratislava, Slovak Republic.

Bajcsy, P., R. Kooper, et al. (2006). Toward hazard aware spaces: Localization using passive RFID technology, National Center for Supercomputing Applications.

Bajcsy, P., R. Kooper, et al. (2006). Toward Hazard Aware Spaces: Knowing Where, When And What Hazards Occur, NCSA.

Bajcsy, P., S. C. Lee, et al. (2006). "Three-dimensional volume reconstruction of extracellular matrix proteins in uveal melanoma from fluorescent confocal laser scanning microscope images." Journal of Microscopy-Oxford 221: 30-45.

Choset, H. M., S. Hutchinson, et al. (2005). Principles of robot motion: theory, algorithms, and implementation, The MIT Press.

Dario, P., M. Bergamasco, et al. (1992). Multiple Sensing for Dexterous End Effectors. Robots with Redundancy: Design, Sensing and Control. A. K. Bajcsy, Springer Verlag.

Dellaert, F., D. Fox, et al. (1999). Monte carlo localization for mobile robots, Institute of Electrical Engineers Inc (IEEE).

Dey, A. K. (1998). "CyberDesk: A Framework for Providing Self-Integrated Context-Aware Services." Knowledge-Based Systems 11 1998: 3-13.

Dishman, E. (2004). "Inventing Wellness Systems for Aging in Place." Computer: 34-41.

East, E. W. (2004). Facility Area Network Workshop.

El-Zabadani, H., S. Helal, et al. (2006). "PerVision: An integrated pervasive computing/computer vision approach to tracking objects in a self-sensing space." issues 10: 11.

Gupta, P. and P. R. Kumar (2000). "The capacity of wireless networks." IEEE Transactions On Information Theory 46 March 2000(2): 388-404.

Hahnel, D., W. Burgard, et al. Mapping and localization with RFID technology.

Hartley, R. I. (1999). "Theory and Practice of Projective Rectification." International Journal of Computer Vision 35 1999(2): 115-127.

Hightower, J. and G. Borriello (2001). "Location Systems for Ubiquitous Computing." IEEE Computer 34 August 2001(8): 57-66.

Hollar, S. E. (2000). COTS Dust, University of California, Berkeley.

Ilyas, M. and I. Mahgoub (2005). Handbook of Sensor Networks: Compact Wirelss and Wired Sensing Systems. Boca Raton, Florida, CRC Press.

Kaddoura, Y., J. King, et al. (2005). Cost-precision tradeoffs in unencumbered floor-based indoor location tracking, Ios Pr Inc.

Kidd, C. D., R. Orr, et al. (1999). The Aware Home: A Living Laboratory for Ubiquitous Computing Research. Second International Workshop on Cooperative Buildings - CoBuild'99.

Lementec, J. C. and P. Bajcsy (2004). Recognition of Arm Gestures Using Multiple Orientation Sensors: Gesture Classification. 7th International IEEE Conference on Intelligent Transportation Systems.

Mainwaring, A., J. Polastre, et al. (2002). Wireless Sensor Networks for Habitat Monitoring. ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02). Atlanta, GA.

Marr, D. (1982). Vision. San Francisco, W.H. Freeman.

Mills, D. (1992). Network Time Protocol (Version 3) Specification, Implementation and Analysis, IETF.

Pankanti, S. and A. K. Jain (1995). "Integrating Vision Modules: Stereo, Shading, Grouping, and Line Labeling." PAMI 17 September 1995(9): 831-842.

Patwari, N., J. N. Ash, et al. (2005). "Locating the Nodes: Cooperative localization in wireless sensor networks." IEEE Signal Processing Magazine 22 2005(4): 54-69.

Priyantha, N. B., H. Balakrishnan, et al. (2003). Anchor-Free Distributed Localization in Sensor Networks, Laboratory for Computer Science, MIT.

Rom'an, M., C. Hess, et al. (2002). "Gaia: A Middleware Infrastructure to Enable Active Spaces." IEEE Pervasive Computing: 74-83.

Roush, W., A. M. Goho, et al. (2003). "10 Emerging Technologies." Technology Review 106 2003(1): 33-46.

Saha, S. and P. Bajcsy (2003). System Design Issues for Applications Using Wireless Sensor Networks, NCSA.

Satyanarayanan, M. (2001). "Pervasive Computing: Vision and Challenges." IEEE Personal Communications.

Scherba, D. and P. Bajcsy (2004). Communication Models for Monitoring Applications Using Wireless Sensor Networks, NCSA.

Scherba, D. and P. Bajcsy (2005). Depth Map Calibration by Stereo and Wireless Sensor Network Fusion. The Eighth International Conference on Information Fusion.

Urban, M. and P. Bajcsy (2005). Fusion of Voice, Gesture, and Human-Computer Interface Controls for Remotely Operated Robot. The Eighth International Conference on Information Fusion.

Urban, M., P. Bajcsy, et al. (2004). Recognition of Arm Gestures Using Multiple Orientation Sensors: Repeatability Assessment. the 7th International IEEE Conference on Intelligent Transportation Systems.

Vildjiounaite, E., E. J. Malm, et al. (2003). Smart Things in a Smart Home. the Fifth International Conference on Ubiquitous Computing: 215-216.

Wang, H., D. Estrin, et al. (2003). "Preprocessing in a Tiered Sensor Network for Habitat Monitoring." EURASIP Journal on Applied Signal Processing: 392-401.

Wechsler, H. (1990). Computational Vision. Boston, Academic Press.

Weiser, M. (1993). "Some computer science issues in ubiquitous computing." Communications of the ACM: 75-84.

Whitehouse, K. (2002). The Design of Calamari: an Ad-hoc Localization System for Sensor Networks, University of California, Berkeley.

Whitehouse, K. and X. Jiang (2004). Calamari: A Localization System for Sensor Networks.

Sustainable Radio Frequency Identification Solutions

Edited by Cristina Turcu

Radio frequency identification (RFID) is a fascinating, fast developing and multidisciplinary domain with emerging technologies and applications. It is characterized by a variety of research topics, analytical methods, models, protocols, design principles and processing software. With a relatively large range of applications, RFID enjoys extensive investor confidence and is poised for growth. A number of RFID applications proposed or already used in technical and scientific fields are described in this book. Sustainable Radio Frequency Identification Solutions comprises 19 chapters written by RFID experts from all over the world. In investigating RFID solutions experts reveal some of the real-life issues and challenges in implementing RFID.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Peter Bajcsy and Rob Kooper (2010). Integration of Data Across Disparate Sensing Systems over Both Time and Space to Design Smart Environments, Sustainable Radio Frequency Identification Solutions, Cristina Turcu (Ed.), ISBN: 978-953-7619-74-9, InTech, Available from: http://www.intechopen.com/books/sustainable-radio-frequency-identification-solutions/integration-of-data-across-disparate-sensing-systems-over-both-time-and-space-to-design-smart-enviro

INTECH
open science | open minds