A Real-Time Wireless Sensor Network for Wheelchair Navigation

J.L. Sevillano, D. Cascado, D. Cagigas, S. Vicente, C.D. Luján, F. Díaz del Rio

Robotics and Computer Technology Lab.
University of Seville, Spain
{sevi, danic, cagigas, satur, cdlujan, fdiaz}@atc.us.es

Abstract— **Today, the availability of inexpensive, low power hardware including CMOS cameras and wireless devices make it possible to deploy a wireless sensor network (WSN) with nodes equipped with cameras for a variety of applications. In this paper, we discuss the use of one of these WSNs as a navigation aid for wheelchairs. Instead of having complicated wheelchairs with lots of on-board sensors, we argue that a viable alternative is to have simpler wheelchairs that are able to interact with an intelligent environment so that the wheelchair bases its navigation on its software intelligence, supported by the information sent by external sensors. Many questions have to be investigated, for instance how sensors should be deployed or whether the wireless links would be able to meet our temporal requirements. We describe some of the solutions we adopted, particularly how to implement with Zigbee devices a polling mechanism that allows us to guarantee a real-time secure navigation.**

Keywords— Real-time communications, Wireless Sensor Networks, IEEE 802.15.4, assisted navigation, intelligent environments.

I. INTRODUCTION

Conventional commercial electric wheelchairs do not provide advanced navigation assistance to help users cope with adverse driving conditions. Onboard sensors are scarce and controllers usually do not allow the programming of high level tasks such as "take me to the desk" or "go straight through the corridor". In research prototypes, a set of sensors and embedded controllers linked together through a real-time communications network are usually found to perform some of these tasks. However, due to the typical design objectives of commercial wheelchairs, low cost components must be used at the expense of loosing precision. Furthermore, using more precise and expensive devices is not useful since typical unstructured environments are intrinsically imprecise and with a high level of uncertainty.

In almost every system described in the literature, both in commercial and research systems (see for instance the review of Ding and Cooper [1]), the sensors used for navigation are incorporated into the wheelchair. Given the recent advances in wireless networks and embedded systems, many of the sensors used to assist navigation may be part of the environment. For instance, ultrasonic sensors in doors or corridors and/or cameras on the ceiling may be used to detect certain types of obstacles. From a more general point of view, the objective would be to take advantage of the environment intelligence to assist wheelchair navigation. This procedure presents several advantages. First, a global view of the whole desired trajectory is available, so that it is possible that some obstacles that cannot be detected locally by the wheelchair may be detected by the smart environment. This includes information related to the environment itself like closed doors, broken elevators, dangerous zones, facilities timetable, etc. This type of information could be permanently updated to reflect any changes in the environment, as well as the appearance of new obstacles. Second, usual problems with mobile sensors like interferences, shaded areas, non-line-of-sight, etc. may be solved using this approach. Since these sensors are fixed, situated in the environment, a careful installation allows them to be positioned and oriented in such a way that these problems are minimized. Finally, the main advantage is that it simplifies the design of the wheelchair itself. Several authors have pointed out that most current prototype wheelchairs are simply too complex to be commercially used [2]. If many of these sensors are situated in the environment (as part of an Ambient Intelligent system), it is enough to have an open wheelchair controller that is able to interoperate with its environment and to receive information relating navigation. Information from the sensors can be used together with the information from the user interface, in what could be called "shared control" [3].

The number and type of sensors depends on the desired level of navigability, and also on the environment. But anyway, several questions arise from the use of external sensors in the wheelchair navigation, that have to do with the transmission through wireless links (connecting the external sensors with the wheelchair) of information with real-time constraints. These constraints cannot be considered hard since

given the number and variety of sensors there will be a certain level of redundancy. Furthermore, the lost or delay of a certain number of packets can be compensated by the subsequent reception of new values from the sensors.

The paper is organized as follows: In section II, we describe our target application, that is, a wheelchair navigation aid based on external sensors. We describe how low cost cameras are used to detect obstacles, how this information may be integrated with user's intentions and how it is sent to the wheelchair through a 802.15.4 wireless link. In section III, we focus on the description of a polling mechanism implemented on top of the 802.15.4 protocol that guarantees the system's temporal constraints. Finally, we present our conclusions.

II. WHEELCHAIR GUIDANCE USING EXTERNAL SENSORS

In this section, we briefly describe the current state of a prototype for a smart wheelchair guidance system. The AmI system includes a Wireless Sensor Network (WSN) that is the base to support indoor navigation of a wheelchair. Zigbee is the wireless technology chosen to support our system. Several authors have pointed out that this standard is well suited for Building and Home Automation, although originally these were not typical applications for sensor networking. Furthermore, home automation is possibly the highest volume shipping application using ZigBee mesh networks today [4]. Our proposal is that an indoor navigation aid system based on Zigbee can benefit from advantages like low power consumption (mainly important for the mobile devices-wheelchair, hand-held devices, etc.), large number of devices, low installation cost, easiness to move and reconfigure public buildings with relatively frequent repartitioning of space (commercial buildings, hospitals, residences), etc.

Our aim is to use sensors that are external to the wheelchair, that is, they are part of the environment. These sensors perform detection and tracking of a mobile object (wheelchair), taking into account possible obstacles. Wheelchair position detection can be obtained from many different positioning systems (for instance, see [5]). Now, we are currently working in a joint project with the Universities of Zaragoza and the Basque Country using advanced positioning systems to locate the wheelchair [6]. However, in this paper we propose to obtain it only from the analysis of video images, using cameras situated on the ceiling. This alternative maybe considered as a low cost version of the system described in [6].

Currently, inexpensive, low power hardware including CMOS cameras are available that make it possible to install a wireless network with nodes equipped with cameras acting as sensors for many different applications. In this way, wheelchair's position and approximate speed, as well as obstacles, can be obtained from the ceiling cameras in order to perform a "local" navigation (avoiding obstacles and allowing reaching specific goals). Furthermore, from the information obtained from the cameras a "global" navigation aid could also be implemented, for instance changing the proposed trajectory in case of crowded corridors, closed doors, dangerous zones, etc. However, this global navigation would probably require the use of information from other environmental sensors, a question that is beyond the aim of this paper. Therefore, in this work we limit ourselves to a local navigation aid.

A. Cameras on the ceiling.

The wheelchair's speed and position are computed by the micro-controller based camera, a CMUcam3 board with a CMOS camera based on an ARM processor (Philips LPC2106). The CMUCAM3 is an open source programmable embedded color vision platform, whose goal is to provide simple vision capabilities to small embedded systems in the form of an intelligent sensor [7].

The hardware architecture consists of three main components: a CMOS camera chip, a frame buffer, and an ARM-compatible microcontroller. The CMUCAM3 can process up to 26 frames per second at maximum resolution of 352 x 288, depending on the application. It has also a MMC interface, two serial UARTs, an expansion port GPIO, software controlled LEDs, and a four-port servo controller. The expansion port on the CMUCAM3 is compatible with various wireless sensor networking motes including the Telos motes. Thus, the CMUCAM3 can be used in wireless sensor networking applications as a smart sensor (see an example in [8]).

The CC3 vision system is the main software for CMUCAM3. The CC3 system is a C API for performing vision and control. The CC3 source code provides a basic image manipulation library and a sample of projects that can be easily extended or modified for custom applications: face detector, edge follow for mobile robots, frame differencing, histogram generation, color tracking, etc. To sum up, the CMUCAM3 targets applications where the use of standard desktop machine would be prohibitive because size, cost or power requirements.

In order to distinguish the wheelchair from other objects, several methods are possible. QR Codes (a two-dimensional bar code) or a simple array of color or infrared LEDs may be used. As a first solution, we use color LEDs which allow us to detect the wheelchair's orientation and make it easier to extend our system to the case of several wheelchairs in the same area by using different combinations for the LEDS (this approach is similar to that described in [9]). The cameras are also used to detect (static) obstacles, distinguishing them from the floor, shadows, etc., using simple, well known image processing algorithms. Note that we are not interested in the research on new image processing algorithms. We prefer to use straightforward solutions that work with off-the-shelves, low-cost hardware.

Since the camera's position is known in advance, the microcontroller of the CMUcam3 board can compute a grid map of the room with cells corresponding to small areas. In our prototype, the resolution can be set between 10 and 20cm. These resolution values are enough for our system, and they keep communication traffic low. These cells may have 2 possible values: "free" and "occupied". The "occupied" cells correspond to obstacles as detected by the image processing algorithms, as well as to the architectural limits defined in a map of the building stored in the wheelchair controller. On the

other hand, the "free" cells represent the space where the wheelchair can move. The wheelchair's position as detected by the camera is also sent as an additional message after the grid map. In this way, only two possible values are required for every cell and thus only 1 bit per cell is needed.

Questions that should be answered are: how many cameras are needed and where these cameras should be positioned. These questions are related to what is known as the coverage problem in sensor networks [10]. Our aim is to ensure that the wheelchair is always within the field of view of at least one camera. Let α be the Angle of View (AOV) of the cameras. Note that this angle depends on the focal length and the image dimensions, and it may be measured horizontally (from the left to right edge of the frame), vertically (from the top to bottom of the frame), or diagonally (from one corner of the frame to its opposite corner).

However, as a first approximation we assume that the AOV of our camera is a constant, that is, it is the same measured horizontally and vertically (this is the case for instance with the camera used in our prototype, as we will see shortly). With cameras situated perpendicularly to the ceiling, the Field of View can be approximated by a circle with radius given by $r = h * tg\left(\frac{\alpha}{2}\right)$, where h is the room height. The coverage problem now becomes the problem of how to arrange identical circles so that they can fully cover the room. Following the reasoning in [11], we take a square with side length a such that the square can be covered by a circle with radius r. In order to do that, the circle must cover two diagonally opposite vertices, so the diagonal should be equal to $2r$. Therefore, $a = \sqrt{2} * r$. Then it is easy to find the number and position of cameras to cover one room, dividing the plane into non-overlapping squares and placing one camera in the center of every square.

For instance, consider a rectangular room with side lengths A and B. The number of cameras (squares with side length a) needed is:

$$n = \left\lceil \frac{A}{a} \right\rceil * \left\lceil \frac{B}{a} \right\rceil$$

We use CMUcam3 with C3088 module equipped with bb218 lenses (α=92' [12]). In a 5x4 meters room with height 3 meters, r=3.11m, and a=4.39m. Therefore, only 2 cameras are needed to fully cover all the room.

B. Shared control of the wheelchair

Once the wheelchair has received the grid map from the cameras (see also section III), free paths and obstacles that should be avoided can be identified. We are interested in a semiautomatic navigation system, so we have to integrate automatic navigation aids with user's intentions. We perform this integration by making obstacles become repulsive virtual potentials, while targets (doors, desks) may become attractive potentials or a mix of attractive/repulsive potentials depending on the situation. For instance, when crossing a door it is useful to make the doorframe become repulsive (see Figure 1).

Virtual Potential Field (VPF) control techniques give us a simple and intuitive way to implement the shared control needed for semiautomatic navigation. It is based on the idea that any obstacle generates a virtual potential field that repels the wheelchair (RVPF-repulsive VPF). To reach the desired goal (e.g. a door or a desk) an attractive VPF (AVPF) is used. VPFs generate virtual forces affecting to the wheelchair and guiding its motion. User intentions, taken from the joystick (or any other steering device), are treated as an additional force by the navigation system for a semiautomatic navigation. "User virtual force" is added to virtual forces generated by VPFs associated to obstacles (occupied cells) and desired goals. Therefore, the resulting force contains both user and automatic navigation intentions. Note that we focus on this "shared control" approach because fully automatic controllers are usually rejected by most users (except for people with very severe mobility restrictions) and furthermore they involve in most cases a very complex and/or expensive system.

The wheelchair controller performs the computation of these VPFs using the information sent by the cameras. For every occupied cell (obstacle) a potential field is computed which is inversely proportional to the distance from the wheelchair. Large objects that occupy several cells are modeled as the sum of these single-cells potential fields.

The information from the cameras can be complemented with other sensors (ultrasounds, positioning systems, etc.). The number and type of sensors used depend on the desired level of accuracy. Information from these heterogeneous multi-sensors should be then combined and integrated.
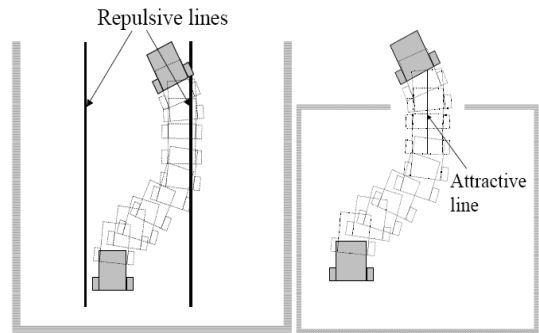

Figure 1. Showing the effects of repulsive and attractive virtual potentials.

There are several sensor fusion approaches available well suited for target tracking applications, like for instance Kalman-filter based methods. In our first prototype version, we use a simple approach consisting in the accumulation, for every cell in the grid map, of the information from the external sensors (an approach that is similar to that of [13]), with the virtual potential field values depending on these accumulated values. For instance, if the accumulated value exceeds a given threshold then the cells are marked as "occupied" (obstacle) or "free".

C. IEEE 802.15.4 as wireless technology

As previously described, the AmI system includes a Wireless Sensor Network (WSN) that is the base to support indoor wheelchair navigation. Zigbee is the wireless technology chosen for our system. This technology provides a low power consumption and low-cost solution, and it also allows us to easily implement a solution that satisfies our temporal requirements, as we will discuss shortly.

All the communication system has been implemented using the CC2431ZDK development kit from Texas Instruments that implements a Zigbee protocol stack compliant with the Zigbee 2006 specification, running in the 2.4GHZ band [14]. The Zigbee Coordinator (see below), connected to one PC for configuration and debugging purposes, has been implemented with a CC2431 evaluation board provided with both, serial and USB interface. Cameras and wheelchair communications have been implemented with CC2430 development boards, supplied by batteries and provided with a serial interface at 3.3V, connected directly to the Camera's serial port or to the wheelchair's serial port through a RS-232 converter (+12V -12V).

In the 802.15.4 standard there are several available configurations [15]. Two different topologies can be used: Star and Mesh (Peer-to-Peer). In the star topology, communication is controlled by a PAN Coordinator, also called *Zigbee Coordinator-ZC*. All devices (up to 255) communicate only with the ZC. In the Peer-to-Peer topology direct communication between devices is possible. Also, two different synchronization modes are possible: beaconless and beacon-enabled. On one hand, the beaconless mode is essentially an unslotted CSMA/CA mode, where nodes can send data to the coordinator at will. In case the coordinator wants to send data to the node, then it must be invited by the node to communicate. No time synchronization occurs in this mode. On the other hand, in the beacon-enabled mode the ZC periodically transmits beacon frames that establish a superframe structure. In this superframe, devices may transmit using a slotted CSMA/CA medium access, with an optional Contention-Free Period-CFP (see Figure 2). Even if these CFP are not used, the periodic transmission of beacons provides a way to synchronize all nodes. Also, it is possible to change the duty cycle to achieve low power consumptions: the device, upon receiving the first beacon, gets current superframe structure and thus know when to active its receiver for the next beacon (just a bit earlier to save power). The device periodically enables its radio to receive the beacon frame, performs the required transmissions and then switches off the radio during the inactive period.

Therefore, generally speaking, when there are energy and/or timing requirements the right choice is the beacon-enabled mode. Furthermore, when any node is mobile (i.e. not fixed) the beaconless mode is not suitable because there is no periodical beacons transmission, so the mobile node may assume its association although it may have lost the link with the coordinator [16]. In the beacon-enabled mode, if a node does not receive a predetermined number of beacon frames then it considers itself as an "orphan" node, beginning a realignment procedure to get associated again with a coordinator.

The beacon-enabled mode is currently limited to Star Topologies, with a network coverage limited to the transmission range of the Zigbee Coordinator, although several attempts try to extend this mode to general Cluster-Tree topologies [17]. For these reasons, in this paper we limit ourselves to a star topology, beacon-enabled configuration.

Since Contention-Free Periods-CFP are not available [14], for this timing-constrained application we needed an alternative way to guarantee that transmissions occur at the desired pace. This mechanism is described in the following section.
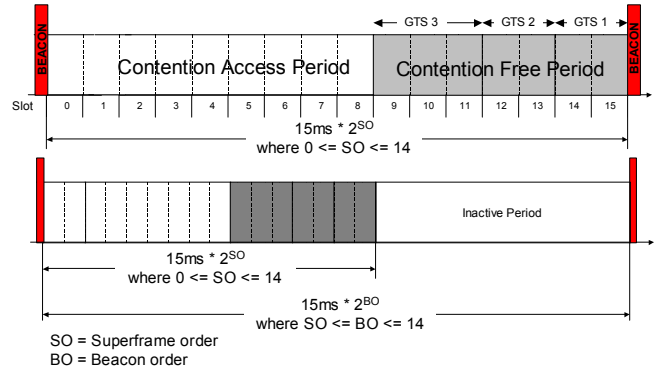

Figure 2. Basic superframe format (upper) and superframe with inactive period (lower).

III. REAL-TIME COMMUNICATIONS

The system obviously has temporal requirements since our objective is the real-time navigation assistance of a wheelchair. For instance, on detection of environmental changes it is required that a timely reaction takes place. A Time-Triggered (TT) architecture is an interesting alternative: it is based on a synchronous design, with each task and message planned a priori in a static schedule. This approach has several advantages: it provides temporal predictability, allows an easy verification of the timing constraints, facilitates fault-tolerance, etc. Furthermore, it makes it easier to implement the periodic measurements required by the multi-sensor fusion approach. From the point of view of a WSN, the power consumption may be reduced since it avoids idle listening, collisions, etc.

A distributed TT architecture needs a common time base shared by all the nodes. For this purpose, a synchronization protocol well suited for WSNs like the Multi-hop Broadcast Synchronization (MBS) protocol [18] may be used. Such a protocol would maintain the nodes (cameras, other sensors, wheelchair) synchronized so that the external sensors can periodically send the grid map to the wheelchair. This solution could be implemented using the beacon-less Zigbee mode, although there are still some open questions to be investigated. However, in our simple prototype (only one-two cameras, static and relatively large obstacles, etc.) the beacon-enabled Zigbee mode may provide synchronization as well as enough bandwidth to support our system, as we will see in the following paragraphs.

Given that a star topology, beacon enabled configuration is used, it is possible to implement an application that takes into account the timing requirements of our system. Briefly, since all communications must go through the Zigbee Coordinator, we let this ZC to manage the order of transmissions. However, several nodes may try to transmit at the same time when a beacon is sent by the ZC, thus causing collisions. The standard tries to avoid these collisions using a slotted CSMA/CA mechanism, waiting a random number of backoff periods previously to any transmission attempt.

However, this mechanism cannot guarantee successful transmissions with upper-bounded delays. Since the active period of a superframe (so called Superframe Duration-SD) is limited to 16 slots, it may occur that this period be insufficient to accommodate all the necessary transmissions (for instance if several nodes want to transmit). In this case, these pending transmissions must wait for the next superframe, being resumed after the next beacon frame. Depending on the duty cycle, the inactive time may cause some timing constraints to be violated. Furthermore, all collisions waste power and bandwidth with useless transmissions.

In order to avoid this situation, we implemented a polling scheme as an alternative approach to share the available bandwidth. Of course, the ZC is the best candidate to perform this polling. The usual way to send data from a coordinator to a given node requires that first the node transmits a data request message. The coordinator then sends an ACK and finally the node is able to send its data. However, this mechanism does not avoid more than one data requests from several nodes, probably causing collisions. A simple solution is to let the ZC transmit a kind of *token* to the selected node, whose address must be listed in the beacon frame. This approach works in such a way that only one node (the one that has previously received this token) is allowed to transmit. This is a common mechanism to share bandwidth when there are temporal constraints [19].

Briefly, the system works as follows (see Figures 3 and 4). It is composed of one Zigbee Coordinator (ZC), a number of cameras on the ceiling (C_1 to C_N) and a wheelchair. We assume that all these devices are situated within the transmission range of the ZC, which periodically transmits beacon frames defining the superframe structure. The ZC includes in every beacon payload the address of the following polled device. This polled device automatically sends its data by means of broadcast packet(s) to the rest of nodes (chairs and cameras). Nodes that don't see their address listed in the beacon frame may still be interested in transmitting their data, but they must wait for the token to be received before they are allowed to transmit.

First, camera C_1 is polled, so it is allowed to transmit the grid map to the ZC with the free areas and the obstacles detected by camera C_1. The grid map may occupy several 802.15.4 frames. For instance, our cameras use C3088 modules equipped with bb218 lenses (α=92'). In rooms with 3 meters height, r=3.11m. If we consider the grid to be a square with side length a, such that the square can be covered by a circle with radius r, then a=4.39m (see section II.II.A). Considering the grid map composed of cells of 10cm (the

highest resolution in our prototype) then the transmission of a binary grid map requires 44x44=1936bits or 242 bytes, that is, three 802.15.4 frames. Note that this resolution is much lower than that provided by our cameras. However, it is enough for our purposes and this way the computational resources needed are kept low.

These frames are transmitted with no competition from other nodes since only the node that received the token is allowed to transmit. This polling policy can be described as an Exhaustive Limited Service Round Robin policy: every node sends all its pending data, with the limit of the superframe duration (more than enough for typical grid sizes).

In the following superframe, the ZC polls camera 2 by sending its address in the next beacon and then camera 2 is allowed to transmit its information to all nodes of the WSN (including the wheelchair/s). The rest of cameras are polled in the same way. Note that with this polling mechanism the duration of the polling period is the same, no matter how many wheelchairs are present in the room.
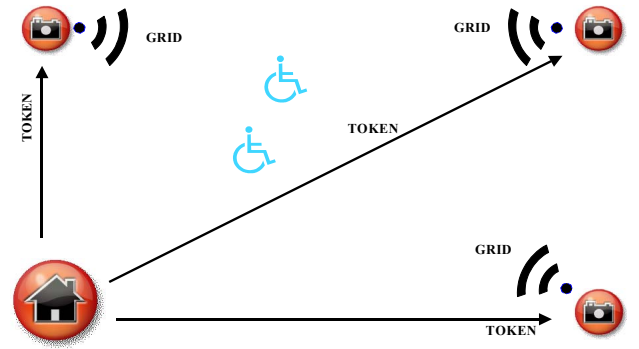

Figure 3.   Message flow between cameras, coordinator and wheelchairs.

The duration of the polling period is $T_{POLL} = Nc * BI$, where N_C is the number of cameras and *BI* is the Beacon Interval. T_{POLL} determines the time between two successive receptions of information from a given camera, so it is the worst case time to detect an obstacle. If we assume that the wheelchair has a maximum speed of V_{MAX}, then the distance the wheelchair travels before it is notified of the presence of a new obstacle is $V_{MAX} * T_{POLL}$. This security parameter determines the Beacon Interval, and therefore the power consumption of communication nodes. For instance, if V_{MAX}=1ms^{-1} and N_C=3, and we assume that a secure distance is 2m, then BI=0.66s (actually, less that or equal to). This is a very hard constraint, since usually a wheelchair is able to stop requiring much less space. In our example, since BI=*aBaseSuperframeDuration**2^{BO}, BO=*macBeaconOrder* and *aBaseSuperframeDuration*=15.36ms (default value) we have

$$BO = \left\lfloor \log_2\left(\frac{666ms}{15.36ms}\right) \right\rfloor = 5$$

So the actual BI in this case is *aBaseSuperframeDuration*$*2^5$=491.52ms. Now we can compute the Duty Cycle as SD/BI=2^{SO-BO}=2^{-5}=0.031. This parameter computes the ratio between activity time and total time, and therefore gives us an idea of power consumption. In this case, only 3.1% out of the whole time the nodes are required to be active.
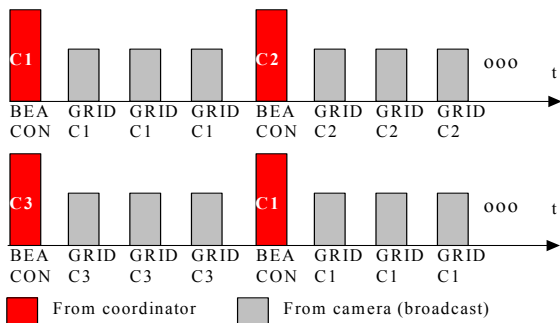

Figure 4. Polling scheme. This time sequence (not at scale) shows how the coordinator requests the grid information from camera 1, 2 and 3 by means of token packets. Once the token is received, only one camera sends its grid information by means of three broadcast messages.

IV. CONCLUSIONS

In this paper, we have tried to demonstrate the applicability of a wireless sensor network (part of an ambient intelligent system) as an assisted navigation aid for wheelchairs. Practical issues like the number and position of sensors (cameras) are discussed. Besides, as one of the most important contributions of the paper, we discuss how the WSN used (based on the IEEE 802.15.4 protocol) can be used in systems with temporal constraints, by means of a polling mechanism. Future work will address the use of a variety of sensors, and how the information from these heterogeneous sensors can be integrated. Also, we are currently working on the use of more advanced mechanisms to synchronize the network, and modeling the system from the point of view of providing temporal guarantees.

V. ACKNOWLEDGMENTS

VI. REFERENCES

[1] D. Ding, R.A. Cooper, Electric-Powered Wheelchairs: A review of current technology and insight into future directions. IEEE Control Systems Magazine. Pp: 22-34, April 2005

[2] T. Dutta and G.R. Fernie, Utilization of Ultrasound Sensors for Anti-Collision Systems of Powered Wheelchairs. IEEE Transactions on Neural Systems and Rehabilitation Engineering 13 (2005) 24-32.

[3] S.P. Parikh et al., Incorporating User Inputs in Motion Planning for a Smart Wheelchair. Proc. of IEEE International Conference on Robotics and Automation. New Orleans, April, 2004.

[4] A. Wheeler, Commercial Applications of Wireless Sensor Networks Using ZigBee. IEEE Communications Magazine, 45-4(2007) 70–77.

[5] Marco et al. Location-Based Services for elderly and disabled people. Computer Communications, 31 (2008) 1055-1066.

[6] J. Abascal, et al. AmbienNet: An Intelligent Environment to Support People with Disabilities and Elderly People. 10th ACM Conference on Computers and Accessibility. Halifax, Canada. October 2008.

[7] The CMUCAM project, http://cmucam.org

[8] A. Rowe, D. Goel, R. Rajkumar. FireFly Mosaic: A Vision-Enabled Wireless Sensor Networking System. Real-Time Systems Symposium (RTSS), December 2007.

[9] I. Fernandez et al. Guidance of a mobile robot using an array of static cameras located in the environment. Auton Robot 23 (2007) 305–324.

[10] C.F. Huang, Y.C. Tseng, A Survey of Solutions to the Coverage Problems in Wireless Sensor Networks. Journal of Internet Technology, 6 (2005) 1-8.

[11] A. Heppes, H. Melissen, Covering a rectangle with equal circles. Periodica Mathematica Hungarica. 34 (1997) 65-81.

[12] http://www.electronics123.com/s.nl/it.A/id.2577/.f?sc=8&category=241.

[13] E. Takeuchi, T. Tsubouchi and S. Yuta, Integration and Synchronization of External Sensor Data for a Mobile Robot. SICE Annual Conf. in Fukui, Japan. Pp. 332-337, August 2003.

[14] Texas Instruments 802.15.4 MAC Application Programming Interface, Version: 1.1. Document Number: F8W-2005-1503. March 19, 2007.

[15] IEEE standard 802.15.4-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). September, 2006.

[16] K. Zen, D. Habibi, A. Rassau and I. Ahmad. Performance Evaluation of IEEE 802.15.4 for Mobile Sensor Networks. 5th IFIP Int. Conf. on Wireless and Optical Communications Networks, WOCN'08. Pp.1 – 5. May 2008.

[17] A. Koubaa, A. Cunha and M. Alves. A Time Division Beacon Scheduling Mechanism for 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks. 19th Euromicro Conf. on Real-Time Systems, ECRTS'07. Pp. 125 – 135. July, 2007.

[18] A. Marco et al. Multi-Hop Synchronization at the Application Layer of Wireless and Satellite Networks. IEEE Globecom 2008. New Orleans, LA.

[19] H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer, 1997.