A Hybrid Approach to Indoor Sensor Area Localization and Coverage

Mengmeng Gai, Azad Azadmanesh, Ali Rezaeian
College of Information Science & Technology
University of Nebraska-Omaha, U.S.
E-mail: {mgai, azad, arezaeian}@unomaha.edu

Abstract — This study presents a hybrid approach to indoor object detection and tracking. A test area is partitioned into regions that are covered by sensors placed on the ceilings. The region location of an object is determined by these sensors using RSSI. The region determination is then followed by the exact localization of the object within the region using Building Information Modeling (BIM) and the 3D stereo image measurements. To determine the coverage ratio of a region for better placement of sensors, the 3D space is partitioned into 2D planes whose coverage points are determined by a newly developed software product called LOCOPac. Other than indoor localization, some simulation results of sensor distribution algorithms that are more suitable for outdoor applications will also be presented. Based on the experimental and simulation results, the proposed approach to indoor localization and LOCOPac has the potential to be used for real world applications.

Index Terms— Wireless Sensor Networks; Building Information Modeling (BIM); Sensor Area Localization (SAL); Stereo Vision; Sensor Area Coverage; Simulation; Network Lifetime.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been used in numerous military and civilian applications such as battlefield, surveillance, biological detection, and environmental monitoring [14,15,31]. What distinguishes these networks from normal ad-hoc networks is that the nodes are small, lightweight, with limited storage and processing power. Two of the fundamental challenges that often arise in sensor network design and deployment are Sensor Area Localization (SAL) [2,4,24] and Sensor Area Coverage (SAC) [9,18,19]. SAL refers to the ability of estimating and computing the physical positions of all or a subset of sensor nodes in a wireless sensor network (WSN), whereas SAC is concerned with the techniques and mechanisms for placement of sensors in a field to quantify how well the sensors sense (monitor) the field. One can easily visualize that the service quality of SAL relies heavily on SAC, i.e. without a decent coverage, localization cannot be performed effectively. Therefore, to provide a unified and manageable approach to the SAL and SAC problems, and at the same time to provide a simulation environment for testing and collecting data, this research has initiated the development of a software package called LOcalization and COverage Package (LOCOPac). The package enables a user to visually observe the movement of an object from one hallway to another on a computer monitor in real time. The package has also the capability to determine the coverage quality of the sensors on a ceiling. To test the approach, a case study is conducted in the Peter Kiewit Institute (PKI) of the University of Nebraska.

Design of sensor networks is not complete without the consideration of important secondary factors such as energy efficiency and fault tolerance [16,19,35,37,39,40]. For this reason, LOCOPac includes the following energy efficiency algorithms: Coverage Configuration Protocol (CCP), Random Independent Sleeping (RIS) and Probing Environment and Adaptive Sleeping (PEAS). To evaluate the performance of these algorithms more realistically, the fault tolerance aspect of sensor node redundancy and random failures of sensors have also been simulated.

Even though the typical applications of sensor localization in real scenarios of WSNs are in 3D terrain, most of the solutions proposed to SAL in the past decades primarily emphasized on two-dimensional (2D) localization [29]. This research proposes a hybrid approach to the 3D SAL challenge by taking advantage of Building Information Modeling (BIM) [25,28] and 3D image measurements. The proposed methodology consists of two phases: region determination (RD) and sensor node localization (SNL). RD attempts at localizing the physical area (region) in which the object resides by deploying a group of sensor nodes with 3D coverage. Once the physical area is determined, SNL obtains the accurate location of the object in the region.

The rest of the paper is organized as follows. Section II provides some general and technical information about the implementation aspects of coverage and localization in 2D and 3D spaces. Section III discusses the methodologies employed in developing LOCOPac. Section IV is devoted to the SAL implementation. Section V presents some simulation examples of SAC. Section VI concludes the paper with some comments for future enhancements.

II. BACKGROUND

This section is partitioned into indoor sensor localization followed by some background information about sensor area coverage.
A. Indoor sensor area localization

The SAL solutions can be classified into two broad categories of centralized and distributed localizations. In the centralized localization approaches [8], sensor nodes receive their positions from a central unit. In particular, the central unit computes the physical location of the sensors based on the localization data collected by the sensors. The physical positions are then sent back to their corresponding sensor nodes. In the distributed localization solutions [26], sensor nodes are capable of handling the relevant computation by themselves, without the support of a central unit. There are already several technologies available for SAL [1,20,21,29]. Some of the contemporary technologies are briefly described below.

Satellite Navigation System (SNS) -- A satellite Navigation System (SNS), e.g., Global Positioning System (GPS), refers to a space-based satellite navigation system which provides location information with critical capabilities to civil, commercial users and military. Because of the line-of-sight requirement, these navigation systems suffer from implementation issues in environments like mountains and indoor scenarios where obstacles may block the satellite signals [21]. In addition, the particular hardware requirements like antenna receivers are critical limitations to the cost factor and power consumption of SNS-based solutions.

Received Signal Strength Indicator (RSSI) -- RSSI-based solutions measure the power of the received radio signal. Based on the received signal strength, distance values can be computed relatively according to the propagation of radio signals. However, measurements based on RSSI are confronted with noises caused by surrounding obstacles and link reliability issues [3,20].

Time of Arrival (TOA)/Time Difference of Arrival (TDOA) -- TOA/TDOA-based measurements compute the range value according to the travel time propagation of a radio signal between a signal transmitter and a receiver. TOA utilizes the absolute travel time between signal units while TDOA takes advantage of the arrival time difference, which can be directly interpreted to a ranging value. This category of solutions can be employed with different signal types such as ultrasonic, acoustic and radio frequency. But interference can cause inaccuracy in TOA/TDOA in some environments [26].

Angle of Arrival (AOA) -- In contrast to the previous approaches that require some form of cooperation and tight timing synchronization between the sender and receiver, AOA has the advantage of locating objects in a stealth and passive manner, which are desirable features for military applications. AOA is based on the capability of sensing the direction of the signal received. AOA requires an antenna array (base stations), which are able to determine the compass direction of the sensor’s signal. The collected information from the base stations is then analyzed to determine the sensor’s location [3]. With the help of antenna array, AOA-based methods can provide reliable measurement accuracy compared to the techniques discussed above. But the drawbacks of this solution are in the additional requirements of hardware equipment and their deployment difficulties.

B. Sensor area coverage

Among a number of metrics used to measure the quality of area coverage, the coverage ratio (CR) is the ratio of covered area to the whole area. Complete area coverage (CAC) occurs if this ratio is 1, i.e. the field is 100% covered. Otherwise, the field is said to be partially covered, which is referred to as partial area coverage (PAC). Another parameter of interest is the rate of sensors deployed per square unit area called sensor density (SD). Specifically, SD is the total number of sensors distributed divided by the total field area. The critical sensor density (CSD) provides an insight into the minimal number of nodes required to reach a desired level of coverage [37].

Another parameter of interest is the failure rate $\lambda$, which is the number of sensors expected to fail per unit of time. As the sensors are often deployed in large quantities, are cheap and have a limited power, the failure rate of sensors becomes a big concern. In LOCOPac, the failure rate is assumed to be exponentially distributed with the cumulative distribution function of:

$$F(t) = 1 - e^{-\lambda t} \quad (1)$$

Deployment Strategies -- The deployment strategies [41] of sensor nodes are concerned with how the placement of sensors in a sensor network is constructed. Two general categories of sensor deployments are static and dynamic deployments. Static deployment means that the placement of sensor nodes before network operation are independent of the network state during the lifetime of the network. The static deployment can further be classified into deterministic and random placements. In deterministic deployment, the sensor locations are preplanned for maximum performance. Random deployment of sensors is another strategy more suitable for remote or hostile fields. In dynamic deployment, placement decision of nodes is based on network changes. For instance, as nodes fail, some nodes might be moved to other locations for better coverage.

Some of the simulation experiments presented in this paper are based on static, random distribution of sensors, using uniform and Poisson functions, which are more appropriate for outdoor applications. Under uniform distribution, $N$ sensor nodes are distributed uniformly over a squared field of size $A$ with sensor density of $\lambda_{SD} = N/A$. In the Poisson model of deployment, the number of nodes deployed follows Poisson distribution with parameter $\lambda_{SP}$. Similar to uniform distribution, each sensor has equal likelihood of being at any location in the region, but the number of sensors to be distributed is Poisson and changes from one deployment to the other.

For indoor applications, deterministic sensor placements are more appropriate. In LOCOPac, the 3D coverage is converted into 2D coverage by simply partitioning the spheres formed by the signal radius of sensors into circles, i.e., slicing the spheres into 2D planes (layers). The 2D coverage simulation can then be used to find the overall 3D coverage by counting the number of pixels that fall into the sliced circles projected onto the 2D planes. To make the explanation more
manageable, the following describes the volume coverage approximation of a sensor that is placed on the center of a room ceiling. Fig. 1 shows a simplistic drawing of the room partitioned into three layers \((L_0, L_1, L_2)\), where the sensor is in layer 0. The sensor has half of its coverage above the ceiling and the other half below the ceiling. As the sensor radius could be higher than the room height, the end cap of the sphere could be above or below the floor.

\[ A = \pi r_c^2 \]

where \( r_c \) and \( h_c \) are the radius of the cap base and the height of the cap from the base, respectively.

To convert the 3D coverage into 2D coverage with good approximation, the area of the circles projected on each layer depends on the number of layers in the room, which is \( h_c/d + 1 \), where \( h_c \) is the room (region) height. Since \( h_c \) is fixed, the accuracy of the volume in (4) depends on how small \( d \) is. The smaller \( d \) is, the more accurate (4) would be. In general, \( d \) defines the resolution of the matrix formed to represent a region. Since the grid approach is used to determine coverage, i.e., whether a pixel is covered by a circle, \( d \) needs to be small enough to provide a good approximation of the region coverage.

C. Energy Efficiency Algorithms

Since sensors have often limited storage of energy, many scheduling algorithms with various operation modes for CAC and PAC have been proposed to control the sleeping patterns of the sensors [37].

Coverage Configuration Protocol (CCP) [39] decides on the sleep pattern of redundant nodes while keeping the coverage at the desired level. CCP is a distributed protocol used for either CAC or PAC. Each node is initially active and runs an eligibility algorithm independently to determine its sleep eligibility. A node \( S \) is eligible to go to sleep if all the points inside its sensing range are already covered by other active nodes in its local neighborhood. Node \( S \) is covered by its neighbors if for each intersection point (crossing point) \( p \) between two sensor nodes \( A \) and \( B \) that is inside of the sensing range \( S \), \( p \) is within the sensing range of a different sensor node \( C \). If so, \( S \) is redundant and thus eligible to go to sleep. CCP can be extended to \( k \)-coverage redundancy easily. A crossing point is \( k \)-covered if it is covered by at least \( k \) distinct sensors. A sensor is \( k \)-covered and thus eligible to go to sleep if all its crossings within its sensing radius with other sensors are at least covered by \( k \) distinct sensors excluding itself.

Random Independent Sleeping (RIS) [16, 37] is a simple algorithm where each node decides independently for its sleep schedule and does not need to know any information from its neighbors such as their locations or distances. RIS can be implemented with various flavors. A simple implementation is to divide the total running time into time slots of equal length \( T_{\text{slot}} \) partitioned into ACTIVE and SLEEP periods. The duration of active and sleep period is \( pT_{\text{slot}} \) and \( T_{\text{slot}} - (pT_{\text{slot}}) \) respectively, where \( 0 \leq p \leq 1 \) is the active probability of a sensor that is globally known by all sensors.

Probing Environment and Adaptive Sleeping (PEAS) [37, 40] extends the network lifetime by inactivating the redundant nodes. A node is always in one of the following states: SLEEP, PROBE, or ACTIVE. Initially all sensors are in the SLEEP state. A node awakes and transits to the PROBE state according to its probing rate. The probe rate provides the frequency with which a node wakes up to decide on becoming active or going back to sleep. Once in the PROBE state, the node broadcasts a beacon signal and waits for a reply from its neighbors. If a reply is received, the node will go back to sleep for another time duration. Otherwise, it will stay active.
III. LOCOPac

LOCOPac, developed in Java, is constituted of two main modules: SAC simulation and SAL implementation. The SAC simulation module contains two components: deployment and simulator. The deployment component contains the deployment algorithms controlling the manner in which the sensors are distributed or the management of sensor activities. The simulator component, based on a chosen deployment algorithm, simulates SAC, either in 2D or 3D space under various scenarios. On the other hand, the SAL implementation module, which is integrated with the SAC simulation module, provides the user interface for the simulation of object localization and more importantly the visualization of object tracking in real time.

The interface is flexible enough to allow for manual intervention of distributing sensors. For instance, a user can move the mouse pointer to insert, move or delete sensors in the field and immediately observe their effect on the area coverage. As an example, Fig. 2 shows the interface for the 3D environment with some sensors distributed by clicking on the 3D on the left side. The right side shows some information about the sensors and their coverage for a specific layer - in this example layer 0 (ceiling). A user can change the layer by adjust the button under “Sensor coverage layers” from left to right. The user can also change the total number of layers. Fig. 3 is another example of 3D environment. In this figure, the field is shown in two different ways, where the left side shows the 3D view of a region in a cube form with sensors on the ceiling. The right half shows the vertical view of the same sensors on a specific layer (layer 0 in this example). The middle bar can be moved to make each partition in the center window larger or smaller. The bottom panel shows some of the inputs, based upon the selections on the right side, and the output results reaching a required accuracy and the coverage attained for the total field.

From the left side of the figure, the user can also choose the 2D option. Choosing this option will treat the field as a 2D plane without any reference to 3D characteristics. The appropriate selection choices appear on the right side that a user can use to set the field size and select the deployment algorithms, and whether the coverage tends to be partial or complete. The selections shown on the right side are in the form of drop-down menu with more details about each selection. The top of the figure shows some general options such as zooming in or out of the field, or moving the field to the right or left for better observation. The bottom panel is used to show the console where the user inputs are shown along with some of the simulation results such as the points covered in a field or the $k$-coverage of the points.

Fig. 4 shows the five choices (in blue) that are so far implemented for 2D that the user can choose from. For example, if the user is concerned with energy efficiency, the simulation results can be affected by the distribution algorithm of the sensors and depending on whether the field coverage is partial or complete.

Fig. 5 shows an instance of SAL user interface, where the center window shows the test space that is partitioned into regions according to the physical structure of the test space. Clicking on the start button under the SAL option will cause the particular menus for SAL on the right side of the figure to appear. These menus include setting the field such as width and height of the field. Below the set field is the interface for SAC simulation, where one can set up the parameters such as sensor size, sensor radius, and the simulation algorithm. There is also a particular interface panel for sensor setting below the simulation panel, which provides the functions to add, remove, or edit sensor nodes in the simulation space.

The real time interface for SAL implementation is provided under the test panel, which includes parameter setting and initializations for the Kinect stereo camera.
Stereo vision technology [10,22] has attracted ample attention and has shown its great potential for ubiquitous sensor localization as it is capable to accurately obtain 3D distance parameters. A stereo vision system is equipped with at least two cameras displaced horizontally from one another in a manner of human binocular vision. Two images from the cameras are compared in order to obtain the depth information of their common points. Generally, several pre-requisite processes are necessary such as camera calibration to remove image distortions and image rectification in order to project two images back to a common plane for computation.

By investigating the principles of stereo vision technique, a Distance Measurement Sensor (DMS) is built with a stereo camera called Kinect manufactured by Microsoft [36,38] and a ZigBee gateway. The DMS behaves as the object to be localized. The purpose of the Kinect is to provide the 3D depth information of the surrounding areas so that the DMS can be localized. The Zigbee gateway is used to communicate with the pre-installed ZigBee sensor nodes in the surrounding areas in order to determine the region location of the DMS.

On the other hand, BIM provides a new perspective on viewing the environment (e.g. a building) in 3D images. It has a wide range of applications on generation and management of virtual representations of physical facilities. In this study, BIM is used to generate 3D spatial information (x, y, and z coordinates) for sensor localization. As shown in Fig. 6, the working process of the proposed methodology primarily consists of three phases: 3D database generation, local distance measurement, and global sensor localization.

In Fig. 6, before the 3D coordinates of PKI\textsuperscript{1} (Fig. 7) can be generated and stored in a database, its image as a 3D BIM model is needed. Generally, the BIM model is chiefly utilized to visualize the design and functional procedures in a 3D view. The PKI second floor is considered as the primary test environment due to its spatial arrangement and sufficient 3D building information. The 3D spatial information (SI) for object localization is then obtained by converting the BIM image to the 3D coordinates. The SI model is generated through sampling every surface of the BIM image via the method proposed by Mark et al. [25] and integrated in the open-source Point Cloud Library (PCL) [30]. In particular, the SI model provides a bunch of 3D spatial points with specific coordinate values (x, y, z).

During the phase of local distance measurement, DMS will measure its 3D distance values from its surrounding working space. Here “local” means the process of distance measurement is implemented based on the coordinate system of DMS. Using the SI model as a reference, the DMS distance measurements will be converted to global coordinates. The relationship between the local and global coordinate system will be discussed in the next section.

The 3D spatial information of the test space is partitioned into sections called regions. For instance a region could be a room in the PKI building. By using the Zigbee-based wireless sensor nodes that are preinstalled in the regions, the responsibility of the global region localization phase is to determine the region where the mobile target node, i.e. DMS, resides. The traditional RSSI-based ranging measurement is used for this

\textsuperscript{1} The realistic 3D BIM model of PKI (Fig. 7) is provided by Kiewit Corporation.
purpose. This information is then combined with the local
distance measurements phase to determine the actual
location of DMS. Recall that RSSI measurements may
not be accurate because of the noises created by the
surrounding obstacles. Therefore, once the approximate
location of DMS is identified, local distance information
will be used to determine the exact location of DMS.

A. The Coordinate Systems

Four levels of coordinate systems are defined, as
shown in Fig. 8a: Geographic Coordinate System (GC) as
the level I coordinate system, the local spatial
information coordinate (LC-SI) system as level II, local
DMS coordinates (LC-DMS) as level III, and the local
coordinates in the Kinect cameras (LC-Kinect) as the
level IV coordinate system. Each coordinate system has
its own origin \( O(0, 0, 0) \) in 3D space. Accordingly, a 3D
point can be described in four different coordinate values,
which can be converted from one coordinate system a to
a corresponding point in a different coordinate system b
using a transformation matrix \( T_{a \rightarrow b} \). For example, Fig. 8b
shows a point in the third coordinate system that is being
converted to its corresponding point in the second
coordinate system using the transformation matrix \( T_{3 \rightarrow 2} \).
Transformation matrices are described later in the
section.

**Level I: Geographic Coordinate System (GC)** – The
GC system refers to the coordinate system that the whole
test space belongs to. Since the global geographic
coordinate values of the PKI building (Fig. 7) are
41.2477208 degree (latitude), -96.0154971 degree
(longitude), and 316 m (altitude), the origin \( O_1(0, 0, 0) \) of
GC system can be actually mapped to this geographic
point. As a result, each point in the PKI building can be
endued with real geographic values on the earth.

**Level II: 3D Local Spatial Information Coordinate
System (LC-SI)** – The local coordinate system in 3D
spatial information model aims at providing the
distance measurements phase to determine the actual
DMS's

**Level III: Local Coordinate System in DMS (LC-DMS)**
-- The DMS local coordinate system serves as the base
coordinate system for the functional implementations of
DMS. LC-DMS system performs differently compared to
the GC and LC-SI systems in that it is mobile while the
other coordinate systems are fixed. Correspondingly, the
process of calculating the transformation matrix related to
this system is also different than the other coordinate
system transformation because of the DMS mobility. In
this scenario, the \( T_{3 \rightarrow x} \), where \( x \) is either 2 or 4, describes
at least two transformations: translation by a distance d
and rotation by an angle \( \theta \) about x, y, or z axis, which are
explained below. These two transformation matrices
along with the spatial relationship between the DMS and
the SI model will be used to determine the DMS’s
position in the GC system.

**Level IV: Local Coordinate System in Kinect (LC-
Kinect)** -- The local coordinate system in the Kinect
provides the pixel positions in 2D images obtained by the
image cameras and 3D depth information corresponding
to the 2D images. Kinect [36] is an RGB-camera
equipped with a horizontal sensor bar connected to a
small motorized base. The base can be positioned
lengthwise above or below the image. The middle point
of the horizontal sensor is the origin coordinate point for
the LC-Kinect system.

The notion of multiple coordinate systems is not new.
For example, multiple coordinate systems are normally
used in robotics to solve different problems. For the
purpose of this study, a separate spatial correlation exists
between two adjacent coordinate systems with the
transformation matrix \( T_{a \rightarrow b} \) that allows the points in
coordinate system a to be converted to their corresponding
neighbor points in a coordinate system b. For this conversion
to take place, one needs to obtain two matrices: translation and rotation. A translation matrix
simply captures the distance between the two sets of
coordinate point values between the two systems, so that
the coordinate values in one system can be adjusted
correctly as they are moved to the other system. Since the
axes of the two coordinate systems may not be in parallel,
the rotation matrix captures the rotation degree along each axis of system $a$ so that it becomes aligned with the corresponding axis in system $b$. As these rotations take place, the coordinate values are adjusted accordingly to keep pace with the rotations.

Fig. 8b shows a 3D point in coordinate system $O_3$ (level III) whose coordinate values are $A: (O_{ax}, O_{ay}, O_{az})$. Fig. 8b also shows the corresponding 3D point $B: (O_{bx}, O_{by}, O_{bz})$ in the coordinate system $O_2$ (level II). Assume, the angles of rotation along each axes $x$, $y$, and $z$ in the coordinate system $O_2$ are $\theta_x$, $\theta_y$, and $\theta_z$, respectively. Further assume that translation along the axes of $x$, $y$, and $z$ are $dx$, $dy$, and $dz$, respectively. Then, the translation matrix $B_1$ and the rotational matrices $B_2$, $B_3$, and $B_4$ along the axes $x$, $y$, and $z$, respectively, are as follows [18]:

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} \cos \theta_x & -\sin \theta_x & 0 & 0 \\ \sin \theta_x & \cos \theta_x & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_3 = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta_y & 0 & \cos \theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_4 = \begin{bmatrix} \cos \theta_z & 0 & 0 & -\sin \theta_z \\ 0 & 1 & 0 & 0 \\ \sin \theta_z & 0 & \cos \theta_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The transformation matrix is then: $T_{a\rightarrow b} = B_2B_3B_4B_1$. Finally, the point $A$ in Fig. 8b will be converted to the corresponding point $B$ by doing:

$$\begin{bmatrix} O_{bx} \\ O_{by} \\ O_{bz} \\ 1 \end{bmatrix} = T_{a\rightarrow b} \begin{bmatrix} O_{ax} \\ O_{ay} \\ O_{az} \\ 1 \end{bmatrix}.$$

B. Region Determination

Fig. 9 shows the test space on the second floor of the PKI building, which consists of a hallway and several rooms. Separate regions in this test space are pre-defined based on the spatial range of the PKI coordinate values. For example, a coordinate point in Region#1 is defined as $x \in \{x_1, \ldots, x_2\}$, $y \in \{y_1, \ldots, y_2\}$, and $z \in \{z_1, \ldots, z_2\}$, where the pairs $\{x_1, x_2\}$, $\{y_1, y_2\}$, and $\{z_1, z_2\}$ are constant threshold values which determine the range values spanned along each axis in Region#1.

As shown in Fig. 6, the overall object global localization receives inputs from Region Determination (RD) and Sensor Node Localization (SNL) to localize DMS. The RD process refers to the estimation of the physical area (region) where the DMS is localized. The implementation of RD process relies on the ZigBee sensors installed in the working space, whose coordinates are pre-known.

The MEMSIC professional kit for WSNs development was utilized to implement the experiments in the region determination phase in order to verify the validation of the developed sensor localization system introduced in the previous sections. The MEMSIC kit is mainly equipped with ZigBee sensor devices, a server gateway, and user interfaces. In particular, the sensor devices are able to perform the low-power wireless sensor network measurements including temperature, humidity, barometric pressure, acceleration and ambient light. In addition to communication with the server (PC) through the USB interface, the gateway is able to collect data from the ZigBee devices. The gateway is also able to gather RSSI values from the sensor devices in order to estimate their corresponding distance relationships from the DMS, so that its region can be determined.

As these rotations take place, the coordinate values are adjusted accordingly to keep pace with the rotations.

The following will introduce the process of RSSI-based distance estimation of sensor devices. Because the gateway attached to the DMS might receive signals from many sensors attached to the test space, and since the RSSI signals used for distance measurement may not be accurate due to the building construction material and obstacles, the selected sensors for region determination are based on their measured signal strengths within a specified range.

Specifically, the signal strength of sensors for various distances to about 4 meters have been obtained and physically validated for their accuracy. MoteView that came with the development kit was used for the measured signal strengths. In particular, a message can be sent by
the gateway asking the sensors to send a reply message to determine their signal strengths. Fig. 11 reflects the relationship between the measured distances and their corresponding RSSI proportional values. For each signal strength received, MoteView provides a corresponding proportional value. Fig. 11 further shows the linear best fit of the RSSI proportional values versus the measured distances, which is \( \text{dis} = -7.3 \times \text{sig} + 460.0 \), where \( \text{dis} \) shows the distance and \( \text{sig} \) is the RSSI proportional value received by the gateway. The approximate range of \((≈0 \ldots ≈60)\) is used as the proportional values to decide which sensors to select. The corresponding distances are then used to determine the DMS region using the trilateration method. The process of obtaining Fig. 11 follows the concept of location fingerprinting [1,7,17], where the received signal strength is compared against a radio map of signal strengths for various locations. The advantage of the radio map, which is constructed off-line, is that the signal strengths measured are customized toward a particular environment. Therefore, more trust can be placed on the accuracy of the measured distances \( \text{dis} \). The other advantage of \( \text{dis} \) is to reduce the false positives caused by sensors from other regions whose signals might reach the other regions.

C. Sensor Node Localization (SNL)

Once the region is determined, SNL is used to localize the DMS accurately. Theoretically, a cloud of \( x-y-z \) point values for a 360-degree view are expected to be entirely matched with the corresponding point values provided by the 3D SI model (Fig. 6). The conversion process from the DMS’s local coordinate system to the digital images pixels that are matched against the SI model is the key in object localization.

The 3D spatial data, i.e. the coordinates, are measured based on the common coordinates of the images with depth information captured by the Kinect camera and the DMS’s coordinate system. The set of points in each image is then compared against the set of coordinates in the 3D SI model. The typical Iterative Closest Point (ICP) algorithm [5] provided by Point Cloud Library is employed to implement the point matching process. Also, the distances are determined using the stereo vision technique [10,22]. Consequently, the coordinates of references in multiple images along with their corresponding distances are combined using trilateration to determine the location of DMS.

Fig. 12 is an example of the 3D point values obtained by DMS. As the DMS moves (Fig. 13), it is able to obtain its accurate position by detecting the coordinate values that are mapped against the corresponding point values of the SI model for the region determined by the pre-installed sensors.

As an example, Fig. 14 shows the overall results of SAL implementation in real time that includes the determined region and the localization of the target sensor node (i.e. DMS) shown as a yellow dot. Recall that accurate localization of DMS requires the assistance of pre-installed ZigBee sensors to determine the region. The gateway installed in DMS keeps detecting the ZigBee sensors whose assigned identification values are mapped against specific regions. Once the region is identified, its borderline (shown in red in Fig. 14) starts flashing. The Kinect attached to DMS is then utilized to measure the 3D depth information between DMS and the surrounding area of the region, which will lead to the precise location of DMS.
Figure 13. Measured distances by DMS in the determined region.

Figure 14. Determined region and localized target sensor.

V. SAC SIMULATION RESULTS

To show the benefits of LOCOPac, this section provides some simulation results of SAC for the uniform distribution and energy efficiency in 2D followed by some examples under 3D coverage showing what the coverage would be by placing the ZigBee sensors on a ceiling of a room. The graphs shown in the following are obtained by using the export option of LOCOPac to export the results to Excel, or by using the internal graph capability option of LOCOPac. Also, although some aspects of fault tolerance are covered in the following examples, the reader is referred to [32] for more details on approaches and simulation results for fault tolerance and security options.

A. Uniform deployment

Under the uniform deployment, Fig. 15 shows the relationship between the number of deployed sensors and CR. The sensing radius is set at 20. The number of deployed sensors is varied from 100 to 1800 in step of 100, and the selected field sizes attempted are 200, 400, 600 and 800 m².

Each coverage ratio in the graph is the average coverage ratio (ACR) of 10 repeated simulation runs for each combination of sensors and field size. According to Fig. 15, Table I shows the SD for at least 80% and 100% coverage. The table shows that SD would need to be doubled to transit from 80% to 100% coverage, which corresponds to doubling the sensors. This transition can be expensive in terms of hardware and energy consumption.

In the next experiment, different number of sensors from 100 to 600 with multiple sensing radii of 10 to 60 m are deployed in a fixed field of 400 m². The simulation is repeated 10 times to get the ACR for the entire field. Fig. 16 shows that when the sensing radius is low, e.g. 10 m, the impact of increasing sensors to achieve high coverage is low. For example, when the radius is 10 m, increasing sensors by 100, i.e. from 300 to 400, from 400 to 500, and from 500 to 600, provides about 8% increase in the coverage and neither change results in high coverage. Looking at this differently, by doubling the sensors from 300 to 600, the increase in coverage is about 24%. Also, the impact of increasing sensors to obtain high coverage beyond 95% is not much. Furthermore, as expected, the coverage ratio increases as the sensor radius is increased to 20 m, but the effect of increasing sensors diminishes beyond 30 m radius.

<table>
<thead>
<tr>
<th>Table I. Relationship between field sizes and their corresponding coverage ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field size</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>200 m²</td>
</tr>
<tr>
<td>400 m²</td>
</tr>
<tr>
<td>600 m²</td>
</tr>
<tr>
<td>800 m²</td>
</tr>
<tr>
<td>1000 m²</td>
</tr>
<tr>
<td>1200 m²</td>
</tr>
<tr>
<td>1800 m²</td>
</tr>
</tbody>
</table>

Figure 15. Relationship between number of sensors and coverage ratio.

Figure 16. Sensor radius vs. average coverage ratio.
B. Energy efficiency algorithms

In random sensor distribution, the number of sensors deployed might be more than what is required to achieve a certain CR. This has the advantage of having areas that are covered by extra (redundant) sensors. In addition, the extra sensors provide for longer network lifetime in case of failures due to the depletion of energy or other forms of faults. On the other hand, coverage of the same areas by multiple sensors wastes energy due to a number of factors such as sensing of the environment or exchanging data with the neighboring sensors. Additionally, the amount of energy consumption is proportional to the data with the neighboring sensors. Additionally, the factors such as sensing of the environment or exchanging by multiple sensors wastes energy due to a number of failures. On the other hand, coverage of the same areas can decrease and still have 100% coverage. A failure rate of 0.05 indicates an average of one failure in 200 units of time. A failure rate of 0.01 is on average a failure in 100 units of time, which is half the running time of 200. This indicates that such failure rates tend to keep the sensor density constant and thus CAC is preserved. However, a failure rate of 0.1, i.e. one failure in 10 units (5% of running time) and higher, tends to be risky as the sensor density can change quickly. Other simulations can be conducted in LOCOPac to see the effect of various parameters on CAC while trying to conserve energy. As an example, simulations have been run to test the effect of failure rates on the total number of sleeping sensors while ensuring 100% coverage, for various k of k-coverage. Because of limited space, the results are not shown, but the simulation results illustrate that the rate of change in the number of sleeping sensors is similar regardless of the value chosen for k.

These graphs show the extent to which sensor density can decrease and still have 100% coverage. A failure rate of 0.005 indicates an average of one failure in 200 units of time. A failure rate of 0.01 is on average a failure in 100 units of time, which is half the running time of 200. This indicates that such failure rates tend to keep the sensor density constant and thus CAC is preserved. However, a failure rate of 0.1, i.e. one failure in 10 units (5% of running time) and higher, tends to be risky as the

The following is an experiment using CCP with complete coverage and with energy efficiency in mind. In the experiment, a static field size of 100 m² is used and 100 sensors with sensing radius of 40 m are uniformly distributed. The coverage redundancy k is set at 1 and the failure rates considered are 0.005, 0.01, 0.05, 0.1, 0.3, and 0.5. For each failure rate, the simulation is repeated with a new sensor distribution until five distributions are found that provide 100% coverage. As the sensors fail, the average sensor density of the five simulations at different points of time is then calculated. Obviously if no failures occur, sensors stay active or sleep for the entire time duration. Therefore, their state changes as failures occur. Fig. 17 plots the results for different failure rates.

These graphs show the extent to which sensor density can decrease and still have 100% coverage. A failure rate of 0.005 indicates an average of one failure in 200 units of time. A failure rate of 0.01 is on average a failure in 100 units of time, which is half the running time of 200. This indicates that such failure rates tend to keep the sensor density constant and thus CAC is preserved. However, a failure rate of 0.1, i.e. one failure in 10 units (5% of running time) and higher, tends to be risky as the sensor density can change quickly. Other simulations can be conducted in LOCOPac to see the effect of various parameters on CAC while trying to conserve energy. As an example, simulations have been run to test the effect of failure rates on the total number of sleeping sensors while ensuring 100% coverage, for various k of k-coverage. Because of limited space, the results are not shown, but the simulation results illustrate that the rate of change in the number of sleeping sensors is similar regardless of the value chosen for k. For further detail, the reader is referred to [32].

Coverage Configuration Protocol (CCP) - CCP is basically a sensor redundancy algorithm that is executed by every sensor to determine its sleep eligibility. Since each sensor determines its fate independently and is only concerned with its own sensing coverage, the sensors have no clue whether the field is completely covered. In other words, the CCP algorithm is a local and decentralized algorithm with the ability to cope with node failures under PAC or CAC.

The following is an experiment using CCP with complete coverage and with energy efficiency in mind. In the experiment, a static field size of 100 m² is used and 100 sensors with sensing radius of 40 m are uniformly distributed. The coverage redundancy k is set at 1 and the failure rates considered are 0.005, 0.01, 0.05, 0.1, 0.3, and 0.5. For each failure rate, the simulation is repeated with a new sensor distribution until five distributions are found that provide 100% coverage. As the sensors fail, the average sensor density of the five simulations at different points of time are then calculated. Obviously if no failures occur, sensors stay active or sleep for the entire time duration. Therefore, their state changes as failures occur. Fig. 17 plots the results for different failure rates.

These graphs show the extent to which sensor density can decrease and still have 100% coverage. A failure rate of 0.005 indicates an average of one failure in 200 units of time. A failure rate of 0.01 is on average a failure in 100 units of time, which is half the running time of 200. This indicates that such failure rates tend to keep the sensor density constant and thus CAC is preserved. However, a failure rate of 0.1, i.e. one failure in 10 units (5% of running time) and higher, tends to be risky as the

The next experiment is focused on partial coverage using the PRIS algorithm. A field size of 200 m² is considered and 100 sensors with sensing radius of 15 m are uniformly distributed in the field. The simulation is run for 200 units of time divided into 5 time slots of 40 units. The simulation is repeated five times with a new distribution for each time. The average CR of the five runs is calculated and compared against the ACR in (5). Fig. 18 conveys that the simulated and mathematical coverage are closely overlapped for the 200 time units of the simulation. This shows that equation (5) indeed shows a good estimate of CR and the fact that LOCOPac provides the correct results. This further shows that the

![Figure 17. Sensor density for various failure rates while keeping CAC.](image)
ACR in (5) can be used to decide on the active probability $p$ to achieve a certain level of coverage.

Figure 18. Simulation vs. mathematical CR in PRIS algorithm.

*Indoor 3D Coverage* - Recall that the region coverage highly depends on the distance $d$ between the layers. Also recall the measured and actual spherical volumes in (2) (3), respectively. Fig. 19 exhibits the simulation results from LOCOPac that shows the accuracy of a single sensor area coverage by increasing the number of layers. In the simulation it is assumed that the region height is 12 units, e.g., 12 meters, and the sensor range is 5 units. A unit is user defined. The accuracy is simply the division of (2) by (3), which produces a value between 0 and 1. Fig. 19a simply shows that as the number of layers increases, implying that $d$ decreases, the accuracy approaches 1. In the figure, when $n = 10$, the accuracy reaches 0.74. With $n = 15$, the accuracy is 0.85. An accuracy of 0.95, as requested by the user, is achieved when $n = 40$. Similar to Fig. 19a, Fig. 19b shows the accuracy of the volume but with respect to $d$. An accuracy of 0.90 requires $d = 0.50$, and accuracy of 0.95, as requested by the user needs the value of $d = 0.31$.

Once the distance $d$ between the layers is determined, the total 3D coverage of sensors can be calculated. Fig. 20a shows the deployment of some sensors in 3D space on the ceiling of a room. The red dots are the sensor nodes, whose signal spheres that are formed project the sensor circles on the layers. One specific plane is marked in yellow (Fig. 20a), and the corresponding sensor circles for the layer are shown in Fig. 20b. The user is able to select different planes. The corresponding output results including the coverage for the selected layer and the total coverage for the field are shown in 21c.

![Figure 19. Volume accuracy of a sensor sphere: (a) Accuracy as the function of number of layers $n$; (b) Accuracy based on distance value $d$.](image)

![Figure 20. Coverage of multiple overlapping circles: (a) Sensor deployment in a 3D cube space; (b) Projected circle formations of a layer; (c). Coverage results produced by LOCOPac.](image)
VI. CONCLUSIONS AND FUTURE WORK

The ability of accurately localizing indoor objects equipped with sensors and cameras in WSNs has been demonstrated by taking advantage of BIM, 3D stereo techniques, and the PCL open source library. The target BIM model of the target space has been converted to a 3D spatial information that is partitioned into specific regions. The region in which a mobile target is localized is determined using a group of pre-installed sensors. The accurate location of the target is then determined based on the image matching algorithm, i.e. ICP algorithm, and the distance measurement technique using stereo image computations. The successful experiments conducted by allowing the object to roam in different connected regions and recording its position at various points of time show that the proposed methodology is potentially useful for real-world applications such as in smart buildings, security, and surveillance.

The detection of moving objects highly depends on the appropriate location of sensors. Since sensors located inside of a building are often installed on places other than floors (to provide better line of sight and minimizing multipath reflection effects), it was necessary to consider 3D coverage of sensors and investigating their effects on coverage depending on their locations. The 3D coverage of a region was obtained by converting the spheres formed by the ZigBee signals into 2D planes (layers). The coverage summation of the planes provided a good estimate of the total 3D coverage. The approach for achieving 3D coverage caused the development and simulation of 2D sensor deployment algorithms, which have the additional advantage for outdoor sensor coverage. In this regard some simulation examples have been presented.

Currently, we are enhancing the indoor coverage and localization in multiple ways:

- **Investigating mobile devices** – The possibility of using smaller mobile devices such as phones and tablets equipped with ZigBee capability are being investigated.

- **Using other network protocol technologies** - Fingerprinting based on WiFi access points instead of using ZigBee sensors is under consideration. WiFi based-fingerprinting has been proposed in a number of studies [12,13,23,33]. Many of these techniques, including this study, use the RSSI absolute values for building the fingerprints. A common drawback among these approaches is the inaccuracy of localization because of the sensitivity of signal strength to multipath fading, types of surrounding obstacles, and device orientation. Additionally, different devices may register different measurements of RSSI, further reducing the localization accuracy. A better approach, as proposed in [6], is to refrain from using the absolute values of RSSI, and instead provide localization based on relative signal strength measurements. In other words, although different devices may provide different RSSIs, their order relation among the measured RSSIs tends to be the same. Consequently, a viable approach to enhance the accuracy of region localization is to use an approach similar to the one proposed in [6].

- **Enhancing LOCOPac** – A number of enhancements can be made to LOCOPac. One is the inclusion of other deployment algorithms such as some of the predefined deterministic algorithms. Other enhancements include rearrangement of options to provide better user interface, or accessing the software via the web interface.

- **Faster processing** – There might be a need for better mathematical schemes to account for faster image processing. Specifically, since the 3D spatial model is generated directly from the BIM model, region determination and accuracy of sensor localization depend on the resolution of the generated SI model. Better accuracy depends on higher resolution, which requires more storage with higher processing time for image processing.

- **Including fault tolerance** - Further investigation is needed for better placement of sensors to ensure complete area coverage while minimizing false positives. We conjecture that this can be achieved using fault tolerance techniques. More specifically, while DMS is receiving signals from the region it is in, it may receive signals from faulty sensors or from sensors in other regions as well, causing the DMS to mistakenly localize its region. To minimize such a problem, we are analyzing and combing multiple signal strengths and removing the outliers to reach a decision with high probability of success.

ACKNOWLEDGEMENT

This research has been partially supported by the NASA Nebraska Space Grant and the Graduate Research and Creative Activity (GRACA) program at the University of Nebraska Omaha (UNO). We also thank the Kiwkit Corporation for providing us with the experimental model of PKI.

REFERENCES


