3DIMENSION LOCATION MANAGEMENT

FOR CONTAINERS

By

SINIL KIM

Bachelor of Engineer

Tongmyoung University

Pusan, Korea

2004

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2008

3DIMENSION LOCATION MANAGEMENT

FOR CONTAINERS

Thesis Approved:

Dr. Venkatesh Sarangan, Thesis Advisor

Dr. Johnson Thomas, Committee Member

Dr. Xiaolin Li, Committee Member

Dr. A. Gordon Emslie

ACKNOWLEDGMENTS

I would like to express my greatest appreciation to my major professor, Dr. Venkatesh Sarangan, for her continued guidance during all phases of my thesis work. His inspiration, encouragement, patience and support helped me in going through this incredible journey. I also thank Dr. Johnson Thomas and Dr. Xiaolin Li, my other committee members. Their valuable effort, assistance, comments and suggestions helped me improve my thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Objectives

The purposes of this study were to find location of containers whose location are unknown to system. This study provided locations information of containers. Such information can assist those maintaining products and keeping from theft.

General statement of wireless sensor network

A Wireless sensor network (WSN) is comprised of large numbers of autonomous sensors which has been developed for use in monitoring environmental and physical conditions, such as sound, light, temperature, sound, pressure and others. An algorithm for a WSN is implicitly a distributed algorithm. Wireless sensor networks are devised originally for military applications. Wireless sensor networks are widely used in many fields such as disaster management and civilian application areas.

Positioning Techniques

There are mainly two algorithms for determining the position of sensors. Range-based localization algorithms calculate the actual distance between a location-known and a location-unknown node. The range-aware algorithms include the convex constraint satisfactions method[4], the DV-based approach[14,19], and MDS-MAP. The range-free localization does not require the actual distance between a location-unknown node and an anchor node whose location is known. DV-based and MDS-MAP is also included in the range-free case.

Nowadays, most of positioning algorithms are relying on one of several forms of raw measurements. These measurements are Received Signal Strength Indication (RSSI), and Time of Arrival (ToA), and Angle of Arrival (AoA).

Received Signal Strength Indication (RSSI) measure the attenuation in radio signal strength between sender and receiver. The power of the radio signal falls off exponentially with distance, and the receiver can measure this attenuation for estimating the distance to the sender.

Time of Arrival (ToA) is based on the speed of radio wave propagation and the measured time it travels between two objects. By using these information, ToA system can estimate the distance between sender and receiver. ToA provides high accuracy rate, but it needs fast processing capabilities to calculate the timing difference for fine-grained measurements.

Angle of Arrival (AoA) techniques allow use of antenna array to measure the angle at which a signal arrives. A major disadvantage of AoA technique requires hardware [1].

Hypotheses

Given the above objectives, it was hypothesized:

1. The size of all containers is the same.

2. The way containers stacked is the same.

3. All sensors are deployed at the same position on a container

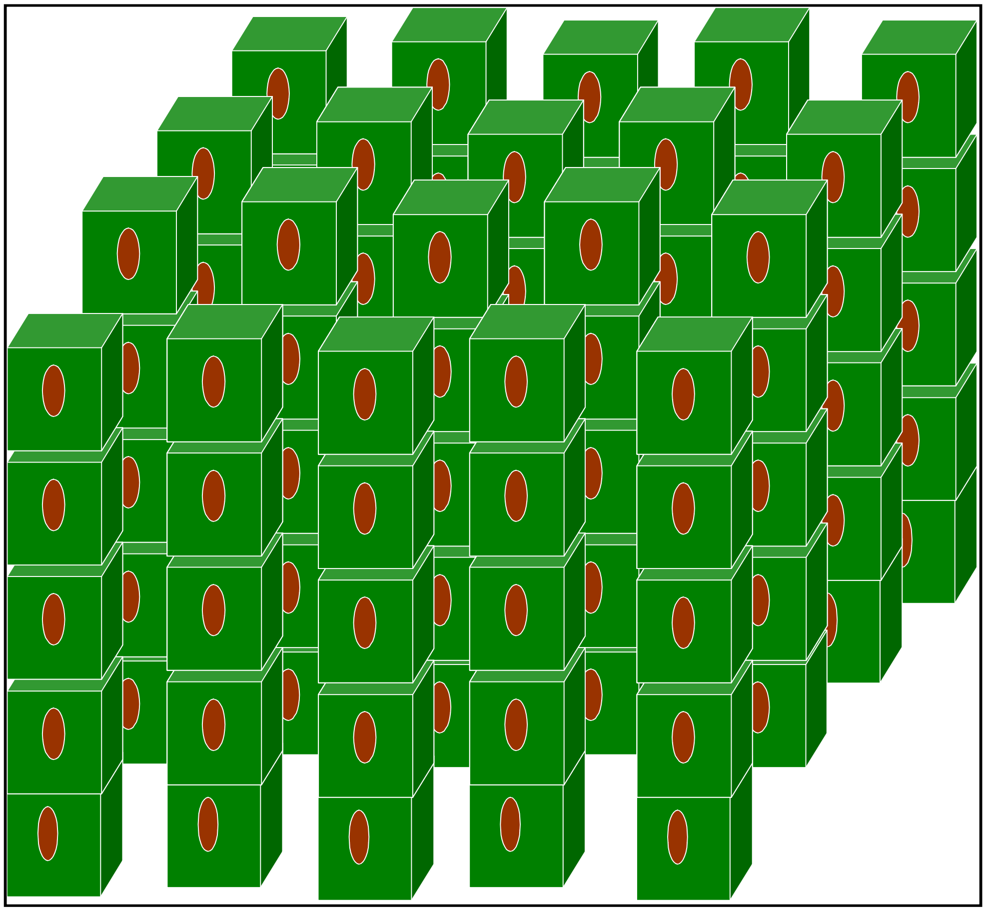

Figure 1. Deployment of sensors on containers

CHAPTER II

REVIEW OF RELATED WORK

Several localization algorithms have been studied for decades. Many positioning techniques use range information, distance information, and anchor nodes of which location are known.

Doherty's [4] convex constraint satisfaction scheme depicts a method for finding location of sensor nodes based only on connectivity. Centralized computation is required for this method. For better performance of this method, anchor nodes to be placed on the outer border, preferable at the corners.

MDS-MAP [22,21] employs connectivity or local distance measure between location-unknown nodes and between location-known node and location-known. Therefore, it leads to better results of localization.

RSSI is used by position schemes in two ways. One is for direct mapping and the other is for indirect mapping. Sorted RSSI quantization [ ] uses indirect mapping. The main difference of other approaches which use indirect mapping is that Sorted RSSI quantization employ a particular quantization model to fairly accurate the mapping. The notion of this method is analogous to that of image of quantization in image processing with the exception of that the quantization is not from incessant RSSI to distinct RSSI. This method initiate with sorting RSSI reading to acquire a sorted range list. Then, it applies a quantizer on the list to create range estimation. In quantization process, range level signifies the range level in a hop.

CHAPTER III

BASIC FS RSSI MECHANISM

In the range-free localization scheme, a node does not require to know its communication range or the real distance between anchor nodes or between an anchor node and a location-unknown node. The actual distance with the actual Received Signal Strength (RSS) is barely applicable to the real wireless sensor network because the Received Signal Strengths varies depending on many different conditions such as weather and power of sender This chapter describes the basic mechanism of the Finding Strongest RSSI. Range-aware scheme calculates the actual distance and should know its distance between anchor nodes or between an anchor node and location-unknown nodes. Disadvantage of this scheme is that if two nodes are in the same range but at different location, system cannot location the position of node. In addition, it hardly locates nodes deployed in 3-Dimension because nodes cannot recognize height of objects.

Location of Anchor nodes and minimum Number of Anchor nodes

To find a location of a sensor on containers, two anchor nodes are needed. One, called Starting Anchor node, is located at (X, Y, Z): 0, 0, and 0 respectively. The other anchor node, called Back Anchor node, is being located at (X, Y, Z): 0, end of Y axis, 0 respectively. In our project, we find a location of sensor in 3Dimention requiring 3 axes in the coordinate: X, Y, and Z. We place 2 anchor and the nodes are located at the bottom of left front most and of left back most, colored in green. That is, Front Anchor node's location of X axis is 0, location of Y axis is 0, and location of Z axis is 0. Back Anchor node's location of X axis is 0, location of Y axis is 3, and location of Z axis is 0. Figure 2 shows where anchor nodes are located. Green color containers represent the containers which contain anchor node.

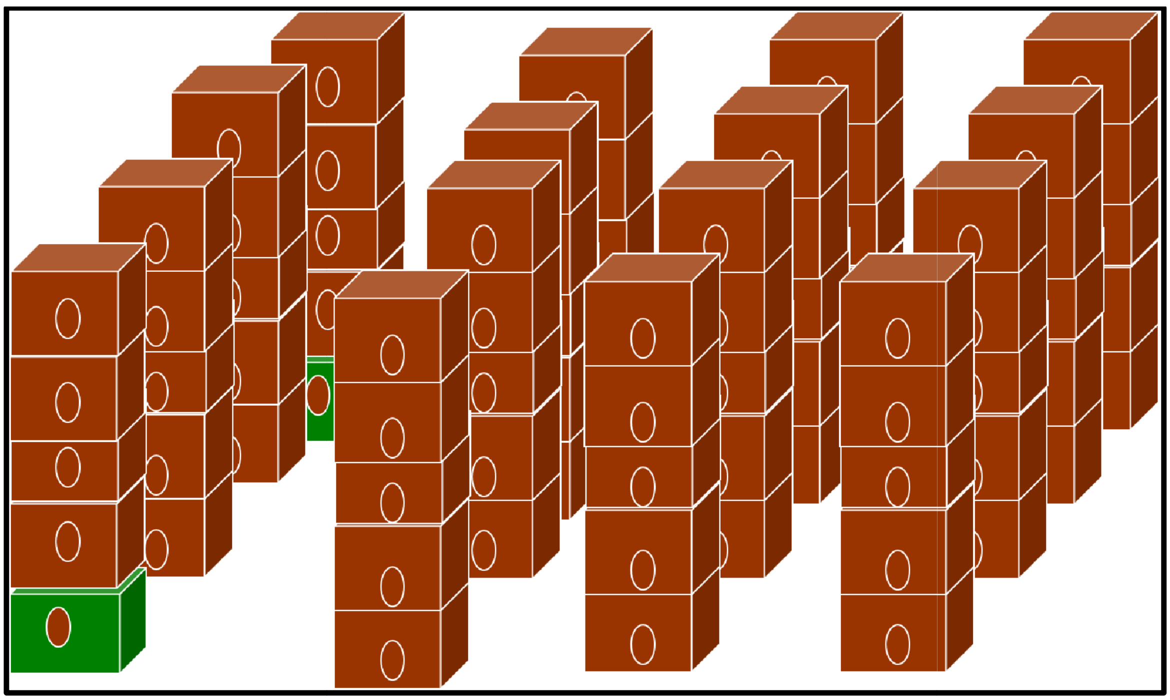

Figure 2. Deployment of anchor nodes

Distance-power relationship

To apply Finding Strongest (FS) RSSI technique to locate sensor's location, we need to know how strong RSSI is. Power of Received Signal Strength is computed as follows

$$\mathbf{P_r = (P_O * d^{-\alpha}) * X \quad (1)}$$

Exponent α is refer is referred to as the distance-power gradient. Alpha α varies according to different environments: For free space $\alpha = 2$ and urban environments, $\alpha = 4$. Variations monitored indoor in is greater than the variation in α monitored outdoors. X is a signal variation. Figure 3 shows the example of distance-power relationship.

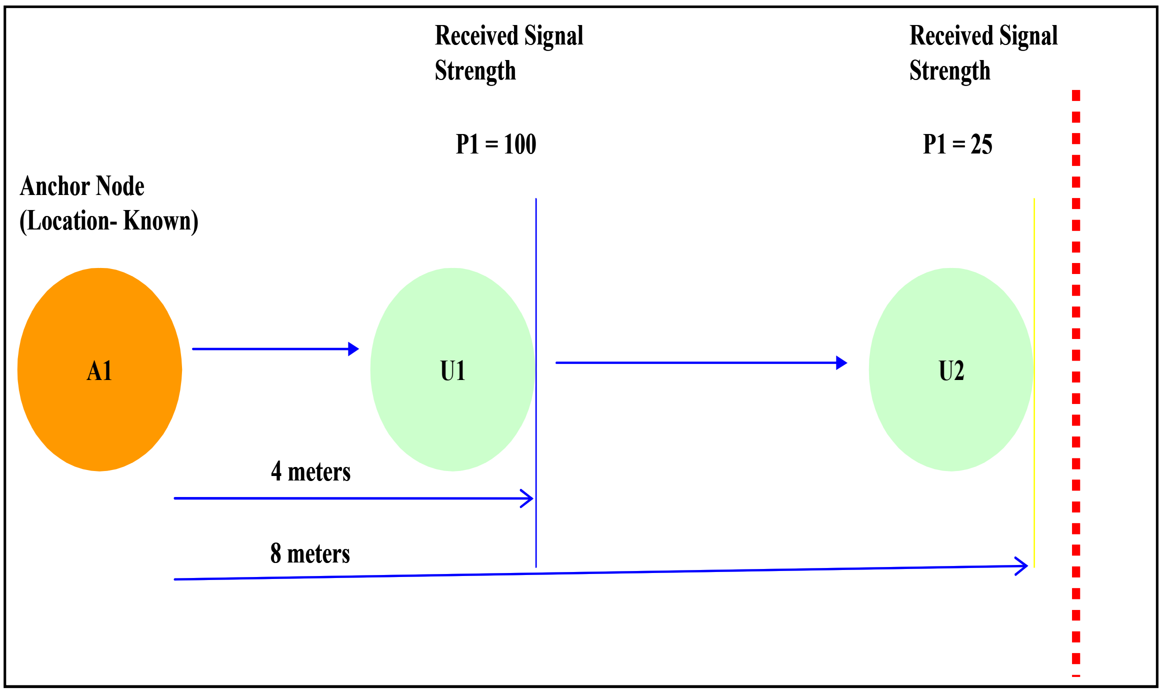

Figure 3. Sending mechanism of Sensor

Mechanism of FS RSSI algorism

As the pictures we showed above, we now know that how many anchor nodes we need, where the anchor nodes are located, and the relationship of distance-power which is calculated by the formula (1). Figure 4 describe the FS RSSI algorism.

[redacted] e
[redacted] not found yet)
[redacted] dcast a RSSI
[redacted] cation table;
[redacted]

Figure 4. Pseudo code of FS RSSI Algorithm for Z axis

First, each anchor node is initialized with “Is_it_bottom = true” and “Is_it_Found = true”. Front Anchor node is initialized with Is_it_Front_node = true, Sending_Node = Front Anchor Node ID, meaning that this node will send first, and bottom_node= Front Anchor Node ID, it tells what node is at the bottom right now. For Back Anchor Node, it is initialized with Is_it_Back_Node = true. Sending node which is Front Anchor Node at the first time, send a signal with RSSI information. Every sensor within the range anchor node’s signal can reach receives the signal. When a node receives a signal from a sending node, every node will calculate logical delay time and wait as much as the delay time. After finishing the wait, a node will send a signal with RSSI information before other node send sends a signal. If a node send signal, other nodes just save both a sending node name and a received node name, and its RSSI information to its own location table. When a node broadcasts RSSI information, it becomes the Node which has the strongest signal. If a node has the strongest received signal strength, location information of X, Y, and Z axis is assigned to the node. X axis and Y axis values are the same as the node which sent a signal. But Z axis is added to one because a node which has strongest received signal strength from a node whose location is found is the sensor on the very upper container due to the distance between the lower container and the upper container. Figure 5 shows how this pseudo code works. After finding a node at (0, 0, 1), it repeat the procedure as the Figure 5 till it finds nodes in wrong location.

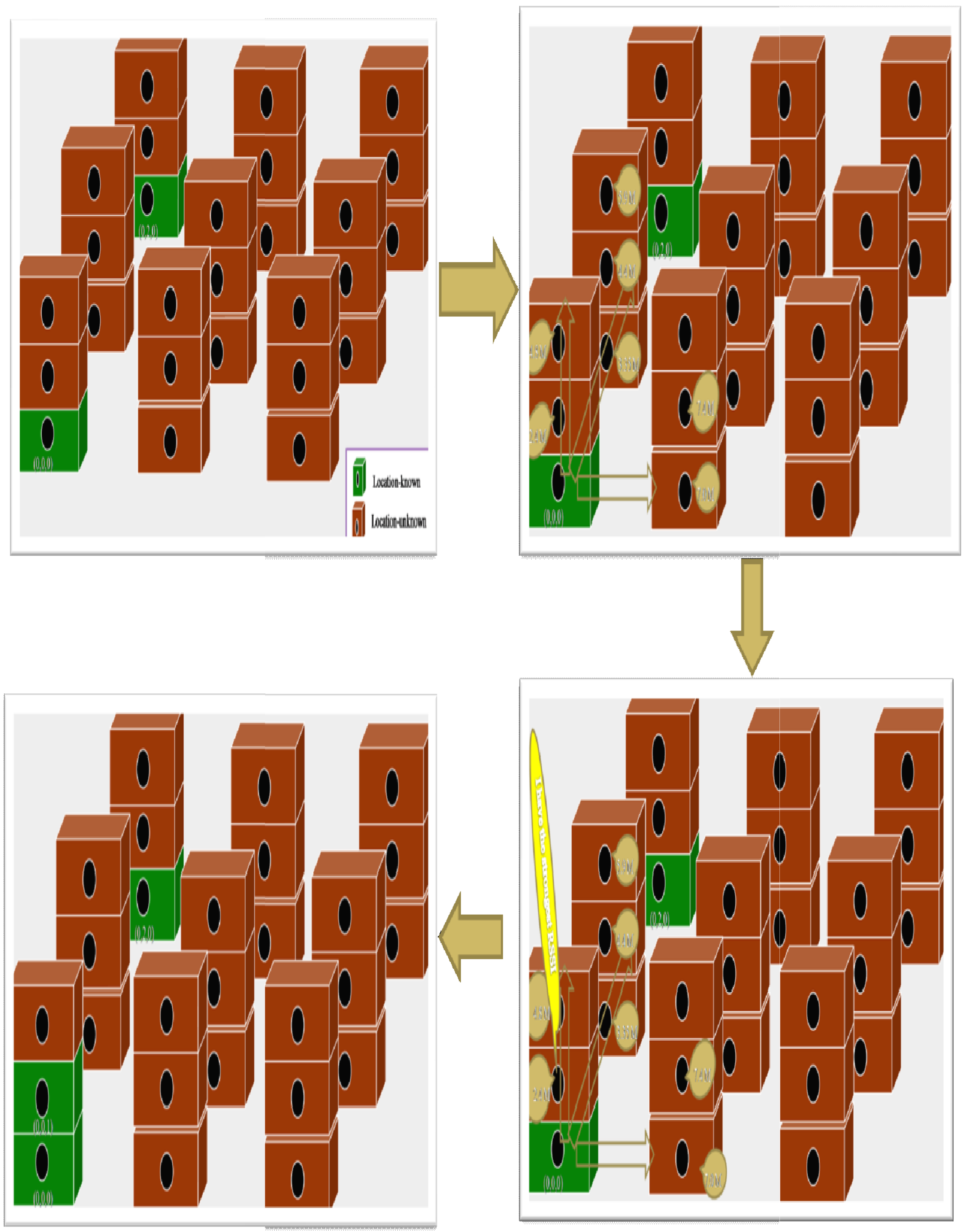

Figure 5. Picture of FS RSSI Algorithm for Z axis

```
If(Strongest_Node. Z axis > = 3)
{
  //RSSI information of found nodes is saved to its own location table
   compare RSSI which received from a node Z-3 to Sending_Node's
RSSI which received from a node Z-3;

   if(Strongest_Node'RSSI < Sending_Node's RSSI)
   {
        Sending_Node = Strongest Node ID;
   }

   if(Strongest_Node'RSSI > Sending_Node's RSSI)
   {
     Strongest.Is_it_found = false;
     Send a signal that makes Sending_Node.Is_it_found = false;
     Sending_Node = bottom;
   }
}
```

Figure 6. Pseudo code of finding nodes has wrong location information

Figure 6 shows that how to check a node with wrong location and correct the wrong location. First, when axis of Z is equal or greater than 3, the strongest node will compare RSSI which received from a node at which is Z-3 to sending node's RSSI which received from a node axis Z-3. For example, in figure 7, a node whose location is (0, 0, 4) compare RSSI that received from a node (0, 0, 1) to RSSI of sending node (0, 0, 3) from (0, 0, 1). If sending node's RSSI is stronger than the strongest node, we assume that the strongest node is at right location. However, in this case, the strongest node has greater RSSI than the sending node. So, we make "Is_it_found " of both the sending node (0, 0, 3) and the strongest (0, 0, 4) false and make next sending node be a bottom node.

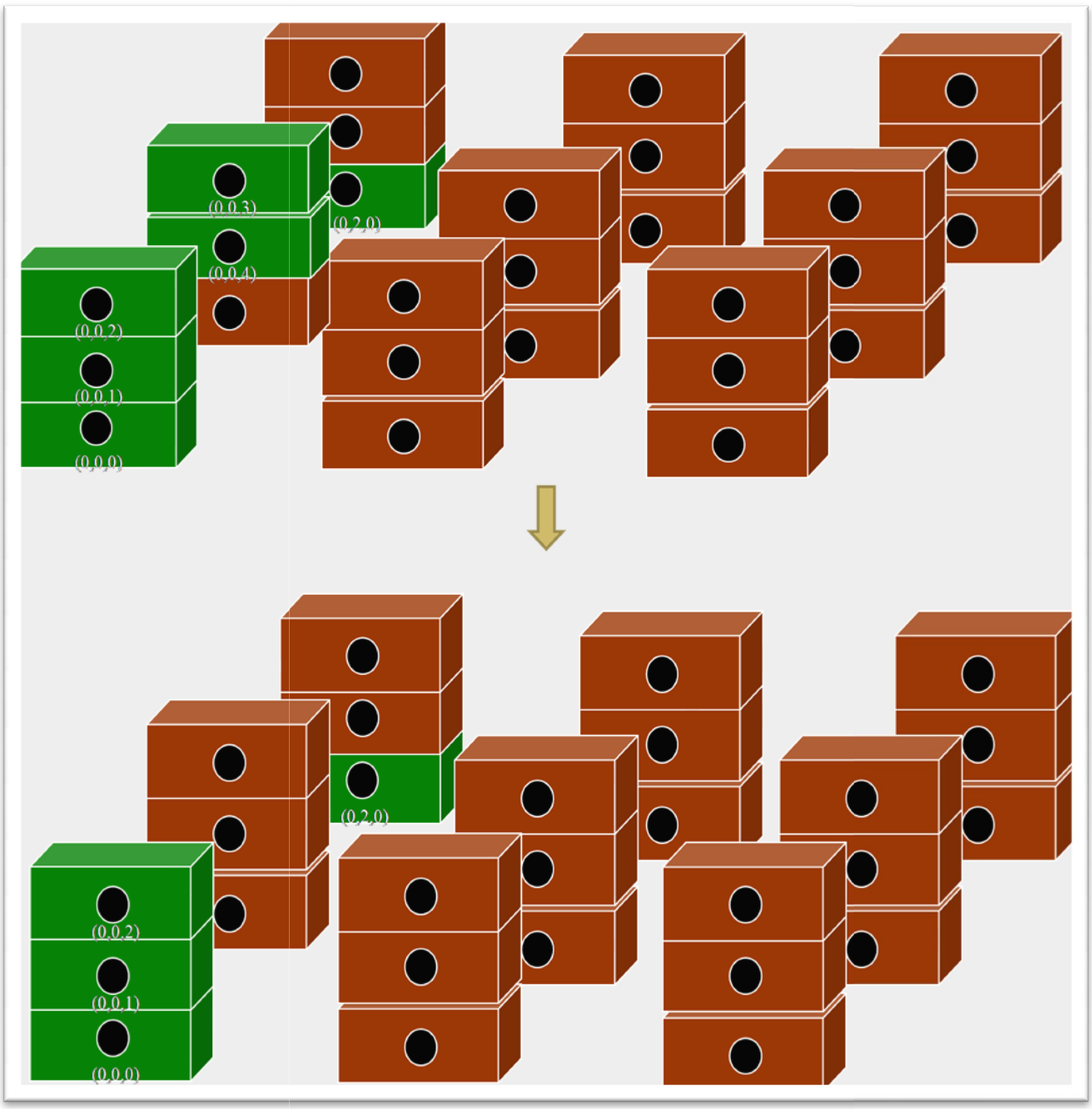

Figure 7. Picture of finding nodes with wrong location

```
Checking Received Signal
                if (a sending_node is already in location table
                    && sending_node  is at the bottom)
                {
                   calculate Delay Time (Pr – RSSI)
                   wait for as much as Delay Time
                 }
                if( a node receive  didn't receive signal from other node )
                   when finishing waiting for the time && the node is not found yet
                   OR if is Back Anchor node
                {
                    Node which has the shortest Delay Time will broadcast a RSSI
                   Strongest     = Node ID
                }
                else
                  Other Nodes save the RSSI information to its own location table;
For Strongest Node
                Strongest_Node .Is_it_bottom = true;
                Strongest_Node. X axis =Sending_Node.X;
                Strongest_Node. Y axis =Sending_Node.Y+1;
                Strongest_Node. Z axis =Sending_Node.Z;
                Sending_Node = Strongest_Node;
```

Figure 8. Pseudo code of FS RSSI Algorithm for Y axis

After finding nodes with wrong location, bottom sensor will send a signal again. Every node within a range that a signal can reach receives a signal as second time. When a node receive signal from the same bottom node second time, this signal is for finding a node in next Y axis. As figure 7 shown, when the bottom node (0, 0 , 0) send a signal, a strongest node is the node in (0, 1, 0) according to RSSI.

```
Checking Received Signal
            if (a sending_node is already in location table
              && sending_node  is Back Anchor node)
            {
              calculate Delay Time (Pr – RSSI)
              wait for as much as Delay Time
             }
            if( a node receive  didn't receive signal from other node )
              when finishing waiting for the time && the node is not found yet)
            {
               Node which has the shortest Delay Time will broadcast a RSSI
              Strongest    = Node ID
            }
            else
              Other Nodes save the RSSI information to its own location table;
For Strongest Node
            Strongest_Node. X axis =Sending_Node.X+1;
            Strongest_Node. Y axis =Sending_Node.Y;
            Strongest_Node. Z axis =Sending_Node.Z;
            Sending_Node = Strongest_Node;
```

Figure 9. Pseudo code of FS RSSI Algorithm for X axis

When Back Anchor node send a signal first time, the node where the node is upper receive the strongest RSSI from Back Anchor node. However, when Back Anchor node send a signal twice, the node will respond is the node located on next X axis as figure 10 shown.

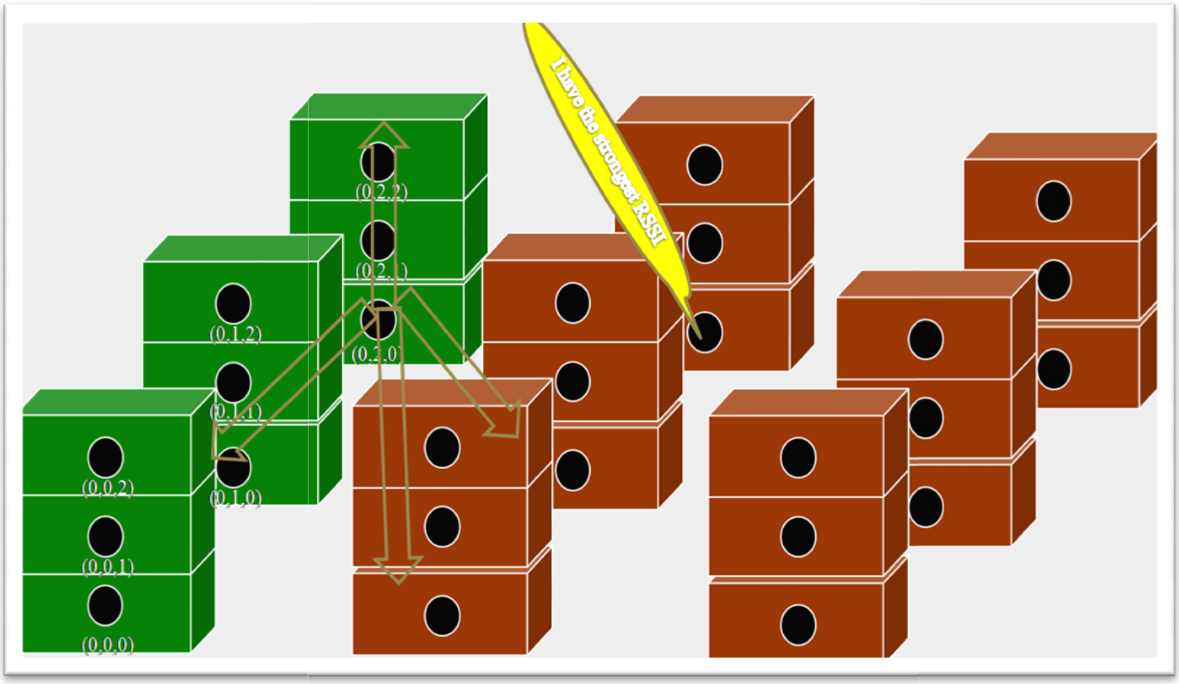

Figure 10. Picture of finding a node in the next X axis

After we find a node in next X axis that has odd number in X axis, has a subtle different way to find location. A major difference between a column in even number of X axis and a column in odd number of X axis is to add 1 to Y axis when a node in even number of X axis or to minus 1 to Y axis when a node in odd number of X axis. Figure 11 show the way to find Y axis. Besides, when a node become (X, 0, 0), the first signal of this node is for a node in Y axis and the second for X axis. This whole FS algorithm repeats till finding entire nodes.

```
Checking Received Signal
                    if (a sending_node is already in location table
                         && sending_node  is at the bottom)
                    {
                        calculate Delay Time (Pr – RSSI)
                        wait for as much as Delay Time
                     }
                    if( a node receive  didn't receive signal from other node )
                        when finishing waiting for the time && the node is not found yet
                        OR if is Back Anchor node
                    {
                         Node which has the shortest Delay Time will broadcast a RSSI
                        Strongest    = Node ID
                    }
                    else
                       Other Nodes save the RSSI information to its own location table;
For Strongest Node
                    Strongest_Node .Is_it_bottom = true;
                    Strongest_Node. X axis =Sending_Node.X;
                    Strongest_Node. Y axis =Sending_Node.Y-1;
                    Strongest_Node. Z axis =Sending_Node.Z;
                    Sending_Node = Strongest_Node;
```

Figure 11. Pseudo code of FS Algorithm for Y axis in odd number of X axis

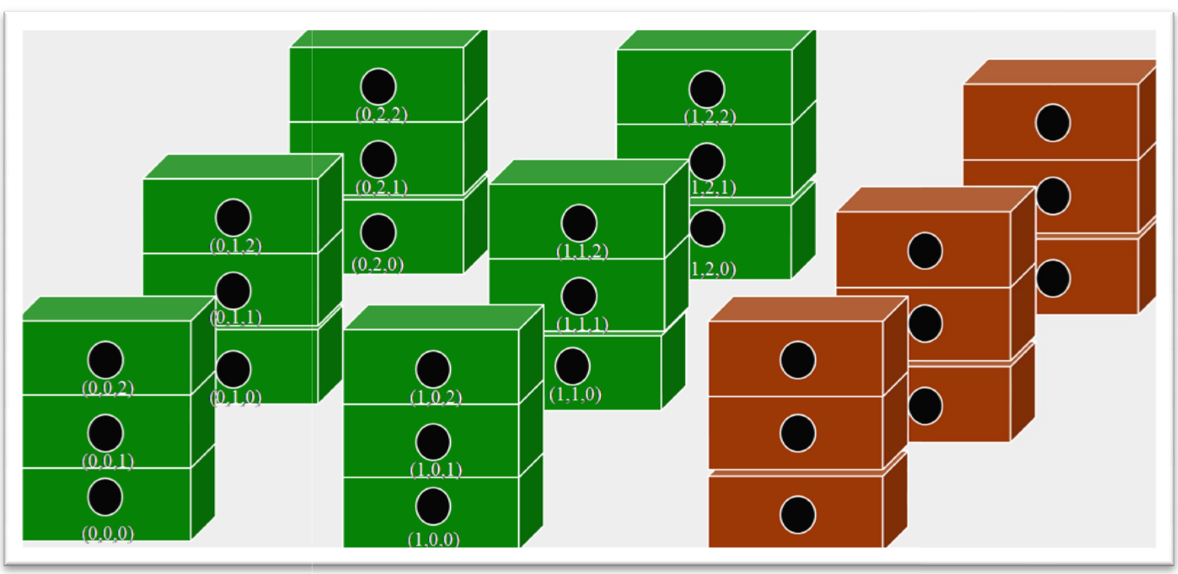

Figure 12. Picture of nodes after finishing nodes at the second column

CHAPTER IV

EXPERIMENT

This chapter presents validation of our algorithm we proposed. We have simulated FS RSSI algorithm in computer programming in JAVA due to the fact TOSSIM which is simulator for wireless sensor does not support 3Dimension.

Environment of experiment

Figure 1 shows a deployment of sensors on containers. Our experiment was not conducted with real environment. We simulated FS RSSI algorithm as if a real environment. We created containers 2^3, 3^3, 4^3, 5^3, 6^3, 7^3, 8^3, 9^3, and 10^3 which are composed of 2 to 10 in each X, Y and Z respectively. There are a few sizes of containers. We selected a container whose length is 6m, whose width 2.4m, and whose height is 2.4m. There is 1m gab between columns of container as you can see the figure 1. We assigned node ID to all sensors, deployed nodes, and assigned all distances between nodes relatively after calculating because nodes are required to know RSSI.

Evaluation

We have conducted experiments by giving different X, an error rate in path loss model for calculating RSSI. We first tested with X = 1.0, X = 1.1, X = 1.2, X = 1.3 and X = 1.4. Each value of X shows different results. We made result graphs according to the

Total number of location corrected nodes, of messages sent, of nodes in right location, and of delayed time.

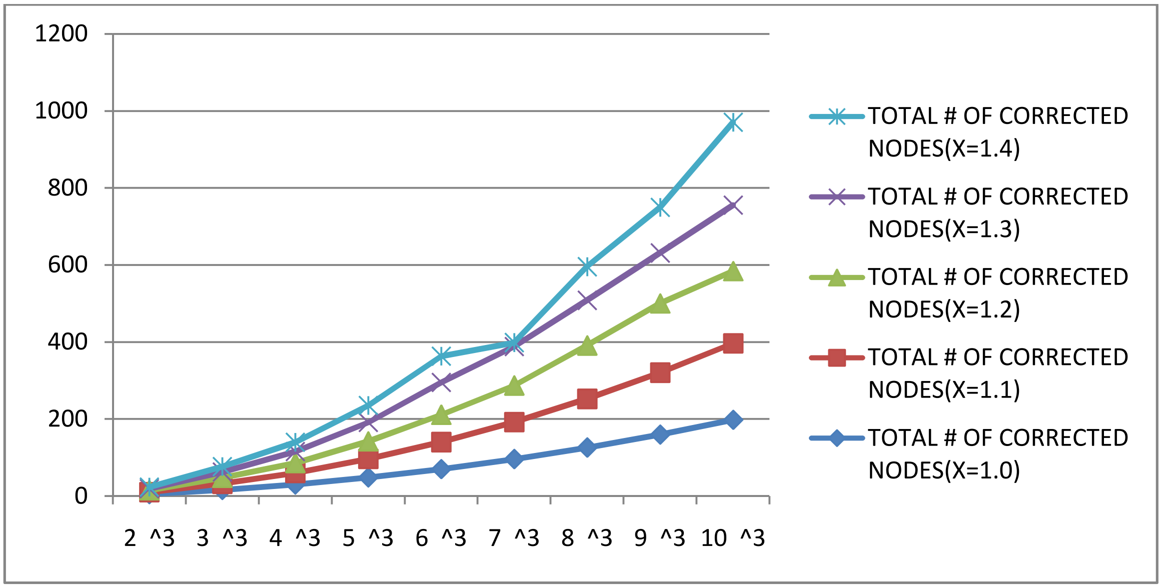

Figure 13. Result Graph of Total number of location corrected nodes

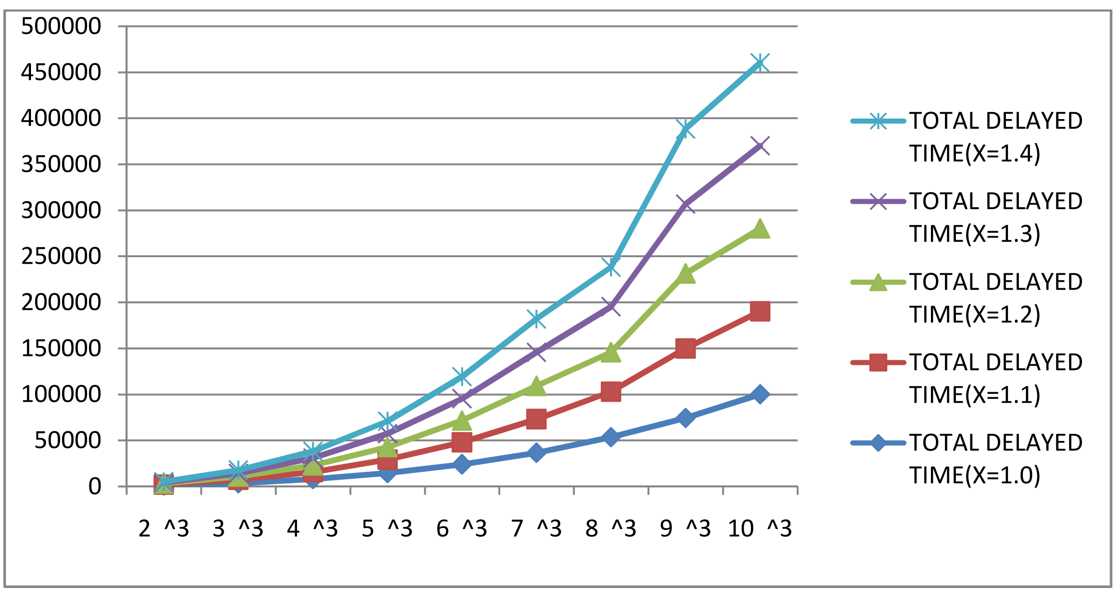

Figure 14. Result Graph of Total delayed time

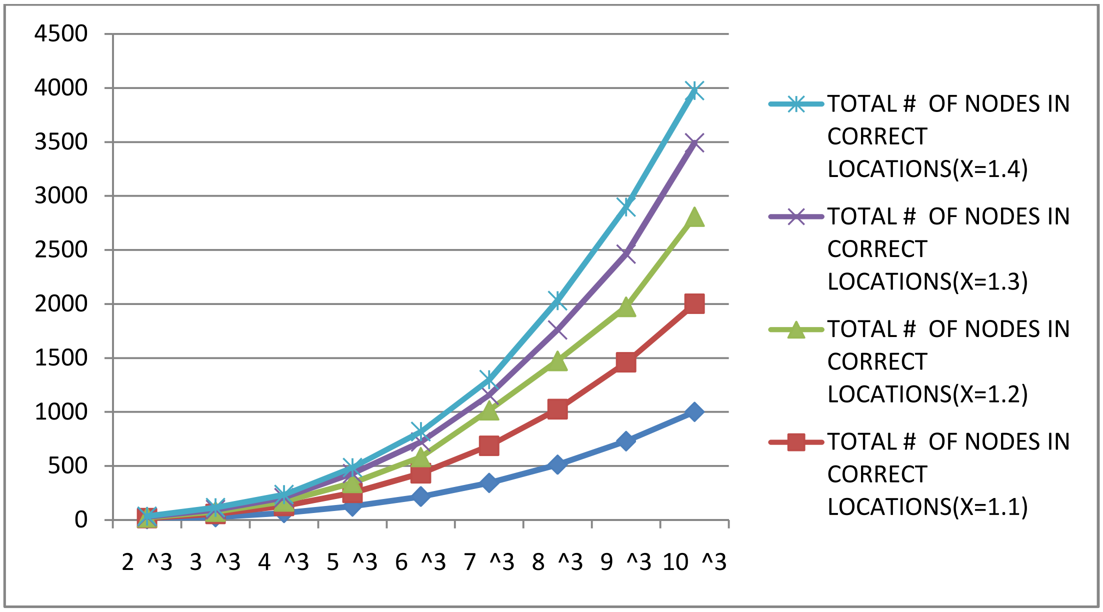

Figure 15. Result Graph of Total number of node with right position

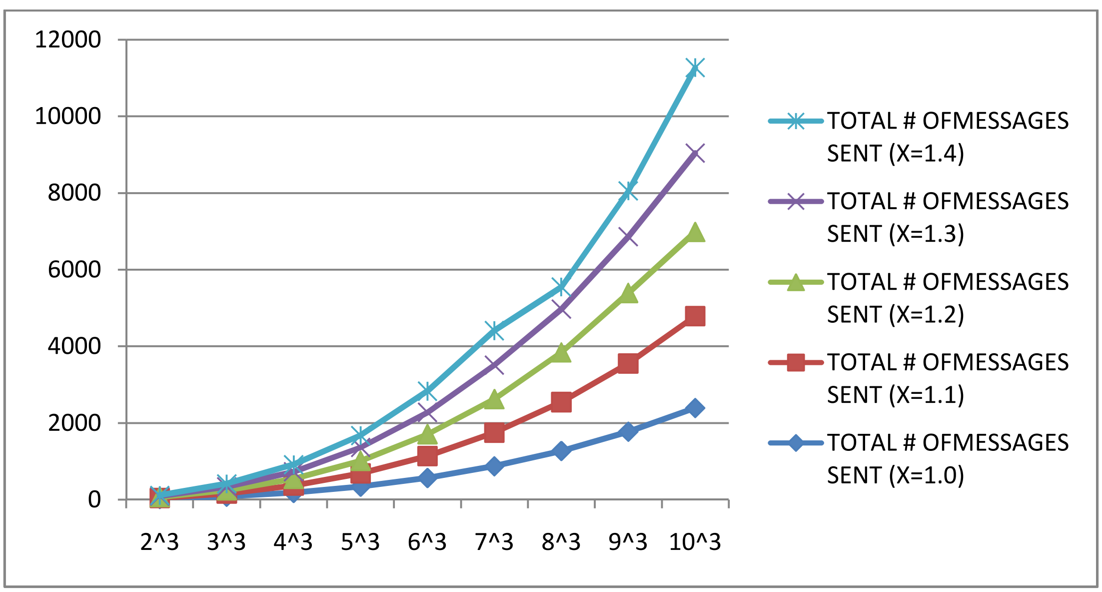

Figure 16. Result Graph of Total number of messages sent

CHAPTER IV

CONCLUSIONS AND FUTURE WORK

In this thesis, we proposed FS RSSI, a localization algorithm to find a location of nodes on containers in 3Dimension. The simulation results shows effectiveness of FS RSSI algorithms and it shows that we are not required to use absolute value of Received Signal Strength. We focused on finding location of containers as well as on minimizing the number of anchor nodes and messages sent.

For the future work we plan to implement FS RSSI algorithm to real sensor which is more reliable and apply the algorithm to different environments which containers are not well stacked and not arranged well.

REFERENCES

[1] Sungwon Yang, Jiyoung Yi & Hojung Cha, “HCRL: A Hop-Count-Ratio based Localization in Wireless Sensor Networks”, in IEEE SECON 2007 proceedings.

[2] Chris Savarese, MS Thesis, “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks”, in department of electrical engineering and computer sciences

[3] http://en.wikipedia.org/wiki/Wireless_sensor_network

[4] Xiaoli Li, Hongchi Shi, and Yi Shang, “A Sorted RSSI Quantization Based Algorithms for Sensor Network Localization”, in ICPADS’05

[5] Kaveh Pahlaven, Prashant Krishnamurthy, “Principles of Wireless Networks: A Unified Approach”, Prentice Hall, 2001

[6] P. Saeedi, D. G. Lowe and P. D. Lawrence, “3D Localization and Tracking in Unknown Environments”, in Proceedings of the 2003 IEEE International Conference on Robotics & Automation

[7] Maurice Walworth, Ajay Mahajan, ”3D POSITION SENSING USING THE DIFFERENCE IN THE TIME-OFFLIGHTS FROM A WAVE SOURCE TO VARIOUS RECIEVERS”, in ICAR ‘97

[8] LIN Dong, LUO Hanwen, SONG Wentao, XU Youyun, LIU Xingzhao, ZHANG Wenjun, “3D Positioning System using Magnetic Field”, in Proceedings of the 5th World Congress on Intelligent Control and Automation.

[9] Yuxin Wang, Han Li, He Guo, Qi Jia, Tianyang Liu, Jun Tang, “A 3D Vehicle Localization Technique in Complex Terrains”, in ISDA'06

[10] Yang Jianxi, Liu Jianting, Shang Zhendong, “Calibrating method and systemic error analysis on binocular 3D position system”, in Proceedings of the IEEE International Conference on Automation and Logistics 2008

[11] Jeffrey Hightower , Gaetano Borriello, Roy Want, “SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength”, UW CSE Technical Report #2000-02-02

[12] Liqiang Zhang, Xiaobo Zhou, Qiang Cheng, “Landscape-3D: A Robust Localization Scheme for Sensor Networks over Complex 3D Terrains”, in 2006 IEEE

[13] Dali Zhao, Yunhui Men, Linliang Zhao, “A Hop-Distance Algorithm for Self-Localization of Wireless Sensor Networks”, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

[14] Yuechun Chu, Aura Ganz, “A UWB-based 3D Location System for Indoor Environments”, in 2005 IEEE

[15] V. Amendolare, D. Cyganski, R. J. Duckworth, S. Makarov, J. Coyne, H. Daempfling, B. Woodacre, “WPI Precision Personnel Locator System: Inertial Navigation Supplementation”, in 2008 IEEE

[16] Maurice Walworth, Ajay Mahajan, “3D POSITION SENSING USING THE DIFFERENCE IN THE TIME-OFFLIGHTS FROM A WAVE SOURCE TO VARIOUS RECIEVERS”, in ICAR ‘97

[17] Shuang Tian, Xinming Zhang, Xinguo Wang, Peng Sun, Haiyang Zhang, “A Selective Anchor Node Localization Algorithm for Wireless Sensor Networks”, in 2007 International Conference on Convergence Information Technology

[18] Ganggang Yu, Fengqi Yu, and Lei Feng, “ AThree Dimensional Localization Algorithm Using a Mobile Anchor Node underWireless Channel”, in 2008 IEEE

[19] Chong Wang, Hongyi Wu, and Nian-Feng Tzeng,” RFID-Based 3-D Positioning Schemes”, *in the IEEE INFOCOM 2007 proceedings.*

[20] Ning Yu, Jiangwen Wan, Qing Song and Yinfeng Wu, ”An Improved DV-Hop Localization Algorithm in Wireless Sensor Networks”, Proceedings of the 2006 IEEE International Conference on Information Acquisition

[21] Doreen A. Thomas and Jia F. Weng, “A Note on Distance-based Geolocation in Wireless Sensor Networks”, Proceedings of ISCIT2005

[22] Shuang Tian, Xinming Zhang, Pengxi Liu, Peng Sun, Xinguo Wang, “A RSSI-based DV-hop Algorithm for Wireless Sensor Networks”, in 2007 IEEE

VITA

Sinil Kim

Candidate for the Degree of

Master of Science

Thesis: 3DIMENSION LOCATION MANAGEMENT FOR CONTAINERS

Major Field: Computer Science

Biographical:

Personal Data: Born in Pusan, Korea, On January 22, 1978

Education: Received Bachelor of Engineer degree in Computer Engineer from Tongmyoung University, Pusan in February 2004. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December, 2008.

Experience: Employed by Oklahoma State University, Department of Computer Science as a graduate research assistant and a graduate teaching assistant from 2007 to present.

Name: Sinil Kim Date of Degree: December, 2008

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: 3DIMENSION LOCATION MANAGEMENT FOR CONTAINERS

Pages in Study: 21 Candidate for the Degree of Master of Science

Major Field: Computer Science

ADVISER'S APPROVAL: Dr. Venkatesh Sarangan