Obstacle-Aware Wireless Video Sensor Network Deployment For 3D Indoor Space Monitoring

Zhonghui Wang

University of Mississippi

Follow this and additional works at: https://egrove.olemiss.edu/etd

Part of the Computer Sciences Commons

Recommended Citation
https://egrove.olemiss.edu/etd/448

This Dissertation is brought to you for free and open access by the Graduate School at eGrove. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.
OBSTACLE-AWARE WIRELESS VIDEO SENSOR NETWORK DEPLOYMENT
FOR 3D INDOOR SPACE MONITORING

A Dissertation
presented in partial fulfillment of requirements
for the degree of Master
in the Computer and Information Science
The University of Mississippi

by
Zhonghui Wang
May 2016
ABSTRACT

In recent years wireless video sensors networks (WVSNs) have emerged as a leading technology for monitoring 3D indoor space in campus, industrial and medical areas as well as other types of environments. In contrast to traditional sensors such as heat or light sensors often considered with omnidirectional sensing range, the sensing range of a video sensor is directional and can be deemed as a pyramid-shape in 3D. Moreover, in an indoor environment, there are often obstacles such as lamp stands or furniture, which introduce additional challenges and further render the deployment solutions for traditional sensors and 2D sensing field inapplicable or incapable of solving the WVSN deployment problem for 3D indoor space monitoring. In this thesis, we take the first attempt to address this by modeling the general problem in a continuous space and strive to minimize the number of required video sensors to cover the given 3D regions. We then convert it into a discrete version by incorporating 3D grids for our discrete model, which can achieve arbitrary approximation precision by adjusting the grid granularity. We also create two strategies for dealing with stationary obstacles existed in the 3D indoor space, namely, Divide and Conquer Detection strategy and Accurate Detection strategy. We propose a greedy heuristic and an enhanced Depth First Search (DFS) algorithm to solve the discrete version problem where the latter, if given enough time can return the optimal solution. We evaluate our solutions with a customized simulator that can emulate the actual WVSN deployment and 3D indoor space coverage. The evaluation results demonstrate that our greedy heuristic can reduce the required video sensors by up to 47% over a baseline algorithm, and our enhanced DFS can achieve an additional reduction of video sensors by up to 25%.
DEDICATION

To my parents, Jun Wang and XiuRong Deng

ZW
ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Feng Wang and Dr. H. Conrad Cunningham, for guidance and patience throughout my graduate study at The University of Mississippi. I would also like to thank Tong Shan, Dr. Jianxia Xue, Tisha Brown for their assistance and advice.

I would like to thank my family for their encouragement and supporting me to pursue a degree in Master of Science.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>RELATED WORK</td>
<td>4</td>
</tr>
<tr>
<td>PROBLEM FORMULATION</td>
<td>7</td>
</tr>
<tr>
<td>STRATEGIES FOR DETECTING OBSTACLES</td>
<td>13</td>
</tr>
<tr>
<td>SOLUTION</td>
<td>17</td>
</tr>
<tr>
<td>PERFORMANCE EVALUATION</td>
<td>22</td>
</tr>
<tr>
<td>CONCLUSION AND FUTURE WORK</td>
<td>35</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>37</td>
</tr>
<tr>
<td>VITA</td>
<td>40</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.1</td>
<td>Facing direction sphere of video sensor with $g_D$</td>
</tr>
<tr>
<td>3.2</td>
<td>Video Sensor Parameters</td>
</tr>
<tr>
<td>3.3</td>
<td>Illustration of dividing 3D indoor space into grids</td>
</tr>
<tr>
<td>4.1</td>
<td>Representation of Lamp Obstacle</td>
</tr>
<tr>
<td>4.2</td>
<td>Representation for Divide and Conquer Detection strategy</td>
</tr>
<tr>
<td>4.3</td>
<td>Intersect point between straight line and rectangular with fixed X axis</td>
</tr>
<tr>
<td>6.1</td>
<td>Number of video sensors as a function of the length of the 3D indoor space</td>
</tr>
<tr>
<td>6.2</td>
<td>Actual simulation of the WVSN deployment for 3D indoor space coverage with obstacles</td>
</tr>
<tr>
<td>6.3</td>
<td>Impact of varying FOVs</td>
</tr>
<tr>
<td>6.4</td>
<td>Impact of varying near fields</td>
</tr>
<tr>
<td>6.5</td>
<td>Impact of varying far fields</td>
</tr>
<tr>
<td>6.6</td>
<td>Granularity of $g_A$ in 3D regions</td>
</tr>
<tr>
<td>6.7</td>
<td>Granularity of $g_D$ for the facing direction sphere</td>
</tr>
<tr>
<td>6.8</td>
<td>Granularity of $g_L$ in 3D regions</td>
</tr>
<tr>
<td>6.9</td>
<td>Impact of varying candidate locations for video sensors</td>
</tr>
<tr>
<td>6.10</td>
<td>Solution for second floor of Computer and Information Science department of University of Mississippi</td>
</tr>
<tr>
<td>6.11</td>
<td>Plane figure of Second floor of Computer and Information Science of University of Mississippi</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Wireless sensor networks (WSNs) have undergone a series of technological advancements. Historically, used for surveillance in government research projects as discussed by Akyildiz et al. (2002), WSNs quickly evolved into useful applications for numerous areas including: industrial, medical, networking, environmental and transportation fields. The wireless sensor network architecture consist of distributed sensor nodes that process, transmit, and receive data efficiently and effectively. As a result, WSNs have expanded into several other categories of sensor node networks. Specifically, in this research we focus on Wireless Video Sensor Networks (WVSNs). Wireless video sensor networks are sensor networks that have cameras mounted on sensor nodes allowing for the recording and processing of images and/or videos as discussed by Soro and Heinzelman (2005). Consequently, in recent years wireless video sensor networks have emerged as a leading technology and popular research topic in academia. Wireless video sensor networks have been widely proposed for deployment in remote coverage areas to monitor large geographical regions (i.e. outdoor spaces) using traditional 2D modeling as discussed by Akyildiz et al. (2002). However, we are focused on the optimal deployment of wireless video sensor networks in a 3D indoor space with the consideration of a more precise obstacle-aware 3D coverage model for video sensors, so that the practical and real world applications can better benefit from this research. Wireless video sensor networks are capable of providing more detailed video information about the sensing field. Additionally, traditional wireless sensors such as heat or light sensors are often modeled with omnidirectional sensing ranges but the sensing range of a video sensor is directional. In particular, as a result of the features of modern camera technology such as Pan,
Tilt and Zoom, the sensing range of a video sensor can be deemed as a fan-shape in 2D and pyramid-shape in 3D. Moreover, in an indoor environment, there are often obstacles such as lamp stands, pendant lamps, or furniture, which introduce additional challenges and further render the deployment solutions for traditional sensors and 2D sensing field inapplicable or incapable of solving the WVSNs deployment problem for 3D indoor space monitoring, and therefore calls for a novel solution.

Extensive research has been conducted in the area of wireless sensor network deployment to provide techniques that optimize sensor coverage and minimize the number of sensor nodes that are deployed, including Huang et al. (2007) and Huang and Tseng (2005). However, the major focuses of these papers only a consideration to the video sensor camera deployment placement negating the impact in which the angular direction of the video sensor, as discussed by Andersen and Tirthapura (2009). Additionally, some papers consider wireless networking but only assume a very simple coverage model with respect to the angle direction such as the work by Tseng et al. (2012). Cardei and Wu (2004), surveyed the fundamental aspects of the sensor coverage problem for WSNs in great detail. To the best of our knowledge, most of the works considering indoor 3D space coverage failed to incorporate an obstacle-aware strategy for obstacles existing in the monitored area which can cause inaccurate results. Specifically, the focus of previous research has been centered around decreasing energy consumption thus maximizing the network lifetime but neglected the unavoidable obstacles. In this research we strive to tackle the deployment problem for 3D indoor space with the consideration of a more precise obstacle-aware 3D coverage model for video sensors.

1.1 Contributions

In this thesis, we study the coverage problem of a 3D indoor space using a Lattice based domain approach that focuses on two specified parameters, a set of candidate locations (position) of the sensor nodes to identify the optimal location and directional angle (orient-
tation) of a video sensor node, while considering the obstacles in the field. We highlight the contributions of our work in this thesis as follows:

- **WVSN Deployment Model for 3D Indoor Space:** To the best of our knowledge, this is the first work to tackle the wireless video sensor network deployment problem for 3D indoor space coverage. We model the general problem in a continuous space, striving to minimize the number of required video sensors to cover the given 3D regions. We then address the problem by converting it into a discrete version where we incorporate 3D grids for our discrete model, which can achieve arbitrary approximation precision by adjusting the grid granularity.

- **Obstacle Awareness:** We designed two strategies to tackle the additional challenges caused by obstacles, which are Divide and Conquer Detection Strategy and Accurate Detection Strategy. These two strategies can help us avoid to cover particles inside the shaded area caused by obstacles when we embed wireless video sensors into 3D indoor space with obstacles. Consequently, we can get very precise and realistic coverage space for each wireless video sensor.

- **Enhanced Depth First Search Algorithm:** We developed the enhanced Depth First Search algorithm that consists of an enhanced searching method to traverse the lattice of local candidate sites for optimal sensor node placement and angular direction. An area coverage function with a greedy heuristic, a derived lower bound for search branch pruning and a simulated frustum culling method are also utilized to increase the efficiency of the algorithm. The performance evaluations demonstrate that our solution can yield high quality results and if given enough time, it can actually return the optimal solution.
CHAPTER 2

RELATED WORK

Over the past decade WSNs have benefit many research areas by providing reliable and scalable technology. As a result, there is now a growing interest in WVSNs as mentioned by Tavli et al. (2012). There are numerous studies available that explored the coverage problem in WSNs, such as the research work done by Raha et al. (2012), Akshay et al. (2010) and Li et al. (2003). However, there are major differences in WSNs and WVSNs, which prevent the usage of techniques that are already well developed for WSNs to be applied in WVSNs, as discussed by Megerian et al. (2005). Introducing WVSNs into an environment presents additional challenges that are not often attributed to WSNs such as coverage quality that depends on the orientation of the video sensor. Another differentiating aspect of the WVSNs is that the sensing range of sensor nodes is directional frustum. In this chapter, we discuss some related studies and explore the distinction in our current work. Chow et al. (2007), used a simple model to provide maximum angle coverage in a VSN to generate a minimum set of sensors to cover all objects of interest. In contrast, our model employs a pan and tilt variable $D$, where the video sensor has vertical and horizontal freedom. Similarly, Sheramin et al. (2010) proposed a Depth First Search algorithm which is implemented for sensing coverage and network connectivity. In their work a sensing disk model is used to formulate the coverage area. All the papers above used simple 2D models in a WSN. As a result, their applications may not be practical in real world settings and thus incur inconsistencies. Two papers similar to ours are written by Kouakou et al. (2012) and Munishwar et al. (2014), respectively. Kouakou et al. (2012) tackled the problem of indoor space target coverage by formulating a k-coverage problem. The proposed heuristic
algorithm ensured k-coverage of the monitored area and ability to determine low cost sensor nodes deployment with obstacles. The maximal cliques generation problem is transformed into the MaxFoV problem for target coverage as discussed by Munishwar et al. (2014) where two polynomial time algorithms reduce the number of candidate FOVs (B-EFA) and find an optimal set of FoVs to be considered by PTZ cameras (G-EFA). A significant limitation for the papers mentioned above is the location of the sensors are deployed for the former paper written by Kouakou et al. (2012), where ours is dynamic, thus removable if a better location is found. Also, there are no cost requirements or constraints associated with the sensor nodes being placed at any point of the 3D space for the latter paper written by Munishwar et al. (2014).

A directional sensor network is implemented by Liang et al. (2011) to maximize the area coverage of a randomly deployed sensor using a greedy algorithm. The objective of the paper written by Wu and Lu (2013) is the target coverage problem for directed sensors with consideration to rotatable angle. However, both works used a simple 2D model and similar greedy approaches to solve the problems. Peng et al. (2013) provide a very detailed 3D practical model of their outlined problem for coverage rate optimization. Using a Greedy Iteration scheduling based algorithm, their solution allows for overlapping of the sensing field of two nodes which can have greater overhead. A novel approach to sensor deployment is considered by TYen (2014) where a PTZ camera is used in a WVSNs. The authors highlighted how a PTZ WVSNs differs from a traditional WVSNs in that there are extended FOV coverages and semi data source nodes. A PTZA heuristic is employed to account for the adjustment time the sensor requires to capture the visual data. The paper most similar to our work is presented by Yen (2013), where a WVSN is used for the sensor coverage problem. A mathematical model is then used to define the problem of complete coverage using a greedy heuristic algorithm FoVIC where the objective is to cover the largest number of uncovered nodes within the area. However, the aforementioned studies were modeled using a 2D approach (i.e. the latter emphasizing a mathematical model) and only used numerical
analysis to evaluate their solutions.
CHAPTER 3

PROBLEM FORMULATION

Wireless video sensors are directional so the coverage area will vary due to the angular direction of the video sensor as mentioned by Soro and Heinzelman (2009). Moreover, based on the hardware features of video sensors, their sensing range is often described as fan-shaped in 2D and cone or pyramid-shaped in 3D as discussed by Akshay et al. (2010) and Munishwar and Abu-Ghazaleh (2013). Therefore, it is pointed out by Tavli et al. (2012) that not only the location of a video sensor affects its coverage area but also the direction can affect the area it covers.

We consider a 3D indoor space (e.g., one floor area of a building), where some 3D regions (e.g., corridors) are required to be fully covered by a number of video sensors. Also, there are some areas such as walls and ceilings that can be used to deploy video sensors. When a video sensor is deployed, its facing direction is also adjusted to some certain position and then does not change anymore. Also within the indoor environment, there are often obstacles (e.g., furniture, lamps) which introduce additional challenges as obstacles can block the line-of-sight of video sensors and reduce their sensing capability. Our goal is thus to optimize the placement and facing direction of each video sensor with the consideration of obstacles, so as to minimize the number of video sensors required to fully cover all the required 3D regions.

3.1 Continuous Space based Model

Given the 3D indoor space, we use $A$ to denote the 3D regions that must be fully covered by the video sensors, $L$ to denote the areas that can be used to deploy the video sensors and $O$ to denote the set of obstacles.
We define a tuple \((C_L, D)\) to denote the location and direction of a video sensor. For a 3D space, the location \(C_L\) can be represented by the 3D coordinate \((x, y, z)\) and the direction \(D = (x', y', z')\) is a point on the surface of a unit sphere (which we call a facing direction sphere) with its radius equal to 1 and centered at \((0, 0, 0)\). We use \textit{face} to denote that the facing direction is the vector from \((0, 0, 0)\) to \((x', y', z')\), as shown in Fig. 3.1. Additionally, we use \(a\) to denote the spatial coordinates (monitoring point) specifically within the 3D regions in \(A\) that we want to cover. \(R_S\) is used to denote the maximum sensing range of the video sensor and we assume that it is the same for all the wireless video sensor nodes in the network. Let \(C(C_L, D, R_S)\) denote the area that a video sensor can cover, which is a function of the location, direction and maximum sensing range of the video sensor. We use \textit{Segment}(C_L a) to denote line segment between \(C_L\) and \(a\), where \(C_L\) is the candidate location of video sensor, and \(a\) is the monitoring point. The video sensors in this research are modeled after a perspective camera as shown in Fig. 3.2 which has static parameters: \textit{far field}, \textit{near field}, \textit{field of view} (FOV) and \textit{aspect ratio}.

Our problem thus can be formulated as to find a set of locations and directions of video sensor nodes \(S = \{(C_{L1}, D_1), (C_{L2}, D_2), \ldots, (C_{Ln}, D_n)\}\), subject to the following constraints:

1. Sensor Location Constraint:

\[
\forall (C_L, D) \in S, \ C_L \in \mathbb{L} ;
\]

2. Region Coverage Constraint:

\[
\forall a \in A, \ \exists (C_L, D) \in S, \ \text{such that} \ a \in C(C_L, D, R_S) \ \text{and} \ \forall O \in O \ \text{Segment}(C_L a) \cap O = \emptyset ;
\]
The meaning of constraint (2) is that monitoring point \( a \) is inside the frustum field covered by \( C_L \), but \( \text{Segment}(C_La) \) does not intersect with any obstacle \( O \in O \). Our objective is to minimize \( |S| = n \).

### 3.2 3D Grid Lattice based Model

Since a continuous space includes infinite number of points, we use a discrete grid lattice based model to approximate the continuous space model, which is also proposed by Soro and Heinzelman (2009). In particular, we divide each region that must be fully covered into discrete 3D grids as shown in Fig. 3.3. As long as all the grid points in the region are covered, we consider the region is fully covered. We use a similar way to divide each area that can be used to deploy the video sensors, where we assume that a node can only be deployed on a grid point within the area. We use \( g_A \) to denote the granularity (i.e., the distance between two neighboring grid points) of the grids used in \( A \) and \( g_L \) to denote the granularity of the grids used in \( L \). In addition, we also divide the surface of the facing direction sphere into grids (like the longitudes and latitudes divide the surface of the earth) and use \( g_D \) to denote the granularity. We assume that a wireless video sensor can only face to a direction where its \( D \) falls on a grid point. To address the scenario of obstacles within the 3D indoor space we introduce two strategies discussed in the following chapter. Fig. 3.3: shows an illustration of dividing 3D cube space into grids and Fig. 3.1: illustrates the facing direction sphere and grid division on its surface. It is easy to see that by adjusting the three granularity parameters, we can easily achieve the required accuracy for approximating the continuous space model.
Figure 3.1. Facing direction sphere of video sensor with $g_D$
Figure 3.2. Video Sensor Parameters
Figure 3.3. Illustration of dividing 3D indoor space into grids
CHAPTER 4

STRATEGIES FOR DETECTING OBSTACLES

To closely model the reality of a real world environment within our 3D indoor space model, we introduce two strategies for detecting obstacle blockage (i.e., to address the line-of-sight blockage of a video sensor), namely, Divide and Conquer Detection and Accurate detection, respectively. In this thesis, we mainly focused on stationary obstacles. We use cuboid to denote regular obstacles in Figure 4.1. 1 We define the single obstacle $O(x_l, x_h, y_l, y_h, z_l, z_h)$ where “$x_l, y_l, z_l$” and “$x_h, y_h, z_h$” represent the low boundary and high boundary in X axis, Y axis, and Z axis for each obstacle respectively.

4.1 Divide and Conquer Detection

For the Divide and Conquer detection we consider two parameters: $C_L$ the location of deployed video sensor and a monitored particle $a$ in $A$. Figure 4.2, Since we know point $C_L$ and point $a$, we first check the middle point $M_0$ between $C_L$ and $a$, if it is in an obstacle then stop. Otherwise, we divide the segment($C_La$) into two subsegments and then check the middle points of these two subsegments, respectively. If any one of the two middle points is in an obstacle, then stop. Otherwise, continue on to divide and conquer, until we have done this for $n$ times and still find no middle points in any obstacles. For the size of $n$, we have to choose it wisely since there is tradeoff between speed and accuracy for the program. If the size of $n$ is too big, even though the detecting accuracy get increased, but it can bring too much overhead to the program. But if the size of $n$ is too small, it can miss detecting some small size of obstacles which leads to inaccuracy result.

1For the obstacles with irregular shape, couple of small cubes can be used to approximate the obstacle with arbitrary accuracy by adjusting the cube size in a similar way to the discussion in the previous chapter.
Figure 4.1. Representation of Lamp Obstacle

Figure 4.2. Representation for Divide and Conquer Detection strategy
4.2 Accurate Detection

The Accurate Detection strategy uses the two points $C_L$ and $a$ to calculate the straight line equation $Segment(C_La)$ in 3D space based on coordinates of these two points. Using $Segment(C_La)$ and obstacle $O \in O$ we check whether there exist an intersection point between $C_L$ and $a$. If an intersection point exists we can conclude that the monitored point $a$ is obstructed by the obstacle within the 3D indoor space. The calculation procedure is shown below: Assume the coordinate of $C_L = (x_1, y_1, z_1)$, and the coordinate of $a = (x_2, y_2, z_2)$, we can get $Segment(C_La)$ as:

\[
\begin{align*}
(x - x_1) &= t(x_2 - x_1) \\
(y - y_1) &= t(y_2 - y_1) \\
(z - z_1) &= t(z_2 - z_1)
\end{align*}
\]

where $t$ is an intermediate value, and $(x, y, z)$ denotes any point on the straight line $(C_L, a)$. As we use cuboid to represent the obstacle, so each obstacle contains six surfaces. The only thing we need to do is to examine that if the straight line $Segment(C_La)$ intersects with one of these six planes of each single obstacle, i.e., if $Segment(C_La) \cap O \neq \emptyset$, the monitoring point is obstructed by the obstacle. Since the obstacle is cuboid, each plane is actually a rectangular. To simplify the problem simple, we assume that all the cuboids used to represent the obstacles are formal, which means that there exists an unchanged axis in the surface coordinate. In other word, all the points on this plane have one same axis value. In figure 4.2, we show an example where X axis dimension is fixed for surface $EFGH$ with value $x_0$, we can then calculate the value of $y_0$ and $z_0$ as:

\[
\begin{align*}
y_0 &= (y_2 - y_1)(x_0 - x_1)/(x_2 - x_1) + y_1 \\
z_0 &= (z_2 - z_1)(x_0 - x_1)/(x_2 - x_1) + z_1
\end{align*}
\]
where \( x_2 \neq x_1 \). We can then examine the following constraints:

\[
\begin{align*}
  & x_1 \geq x_0 \geq x_2, \; x_1 \geq x_2 \\
  & y_h \geq y_0 \geq y_l, \; y_h \geq y_l \\
  & z_h \geq z_0 \geq z_l, \; z_h \geq z_l
\end{align*}
\]

If all these constraints are satisfied, then the straight line \( Segment(C_L a) \) intersects with plane \( EFGH \) at point \( I \), which means that point \( a \) is obstructed by the obstacle. When \( x_2 = x_1 \), either the straight line \( Segment(C_L a) \) is in parallel with plane \( EFGH \) (if \( x_0 \neq x_1 \)) or part of the straight line \( Segment(C_L a) \) is inside plane \( EFGH \) (if \( x_0 = x_1 \)), where the latter case also indicates that point \( a \) is obstructed by the obstacle. This strategy, guarantees that our result is correct. But this strategy is little bit slower than the Divide and Conquer Detection strategy since more calculations are needed to accomplish it. We can use this strategy to check when the size of obstacle is smaller. We incorporated the Accurate Detection strategy into both Greedy Algorithm and Enhanced DFS Algorithm in the next chapter.
CHAPTER 5

SOLUTION

Recall that the objective of our problem is to find a set of locations and directions for wireless video sensor nodes to fulfill the corresponding location and coverage constraints while minimizing the total number of required video sensors. In this section, we will tackle this problem by starting from a greedy heuristic algorithm. We will then propose an enhanced DFS algorithm with pruning, which can yield high quality results efficiently and if given enough time, can actually find the optimal solution.

5.1 Greedy Heuristic Algorithm

Algorithm 1 illustrates the greedy heuristic algorithm we have designed. The objective of the greedy heuristic algorithm is to achieve complete area coverage of the 3D regions $A$ by determining the candidate locations $C_L$ and directional angle $D$ to cover the maximum number of lattice points, where each point is covered by at least one video sensor. A while loop is implemented to check if there are coverable 3D regions and video sensor placement locations available to continue (line 1). Within the $For$ loop, we loop through each candidate location $C_L$ in $L$. For each $face$, we check each monitoring point which is within the covered field by current candidate location $C_L$ to see whether it is obstructed by $O$ or not. We compute all the $face$ for the candidate locations $C_L$ and pick up the $face$ which covers the maximum number of monitoring points $F$(line 4). After the $For$ loop, we record the $C_L$ which cover the maximum number of monitoring points in $L$. Instead of arbitrarily choosing a location in $L$ to deploy a video sensor, in each iteration of the $while$ loop, the greedy heuristic algorithm strives to choose among the candidate locations in $L$ and find the location where the deployed video sensor can cover the maximum number of lattice points $F$ (lines 9). Note
Algorithm 1 Greedy Heuristic Algorithm

\textbf{Input:} $A = \emptyset$, $D = \emptyset$, $L = \emptyset$

\textbf{Output:} Set of $S$ with max $F$ in descending order

\textit{Initialize :} list of $A$, $D$, $L$ and $final$.

1: \textbf{while} $L > 0$ and $A > 0$ \textbf{do}
2: \quad \textbf{for} each $C_L$ in $L$ \textbf{do}
3: \quad \quad Compute all $face \in D$ for $C_L$ with consideration of obstacles.
4: \quad \quad Select $face$ with max $F$.
5: \quad \quad \textbf{if} $(C_L(F^*) > C_L(F))$ \textbf{then}
6: \quad \quad \quad Update $C_L$;
7: \quad \quad \quad \textbf{end if}
8: \quad \textbf{end for}
9: \quad record $C_L$ with max $F$ into $final$
10: \quad remove $C_L$ from $L$ and update $A$
11: \textbf{end while}
12: \textbf{return} $final$

that after a location is chosen by the greedy heuristic algorithm, since a video sensor is deployed at that location and covers a number of fresh points which has not been covered by any video sensor, the number of fresh points remained to be covered by each remaining candidate location in $L$ needs to be recalculated, and the one that maximizes the number of covered fresh points after recalculation will then be chosen as the next location to deploy a video sensor. A list of $S = \{(L, D)\}$ that will completely cover the 3D region is then returned (line 10).

5.2 Enhanced DFS Algorithm with Pruning

We design an enhanced DFS algorithm to further improve the quality of our solution, which given enough time can actually return the optimal solution. In a standard Depth First Search algorithm, we need to explore each search branch that picks a location in $L$ and a facing direction. Since the solution space can expand quickly with the size of $L$, this makes the algorithm very inefficient. In our design, we use pruning which can cut off most of the solution space and thus significantly improve the efficiency of our enhanced DFS algorithm, as presented in Algorithm 2. To achieve this, the first enhancement is that instead of starting
the search from scratch, we use the result of our greedy heuristic algorithm as the currently found best solution\(^1\), so that all the search branches (depth) that have already used equal or more number of video sensors compared to the currently found best solution can be safely pruned (lines 1-4). The second enhancement is the selection of the location in each search step, instead of choosing the next location by the default order, we sort all the candidate locations by the decreasing order based on the maximum number of fresh points that a video sensor at these locations can cover and then choose by the sorted order (lines 8-13). This approach allows our algorithm to quickly find high quality solutions and skip as many low quality solutions as possible. Also, we apply a similar enhancement when we choose the facing direction of a video sensor. Another enhancement is that we derive a tight lower bound to estimate the number of video sensors that we still need to deploy to cover all the remaining fresh points (lines 17-19). A recursive call to the enhanced DFS search is implemented in line 22. The lower bound is calculated based on the sorted candidate locations and directions in the previous two enhancements, where we keep choosing the location and direction from the front of the sorted results and add the number of fresh points covered by the chosen location and direction together until the sum is equal to or greater than the total number of fresh points that actually need to be covered. The lower bound for the enhanced DFS algorithm is defined by the following constraint:

\[
\sum_{i=1}^{n} (C_L, D)^i_{F_c} \geq F_R, \text{ and } n + \text{depth} \geq \text{best}
\]  

(5.1)

where \(n\) is incremented by one until this constraint get satisfied. In constraint 5.1, if the estimated number of new candidate location \(n\) plus the number of video sensors that we have already deployed is equal to or greater than the currently found best solution, the search branch can be safely pruned (line 18). When a search branch is cut off or fully explored the search will revert to its previous status. A minimized set of \(|S| = n\), where \(S\) is optimal is

\(^1\)From the greedy algorithm, we calculate a number of cameras for covering. We use this number as our baseline for the enhanced DFS algorithm. At the first iteration, the “best” term equals to the baseline.
Theorem 1. *Given enough time, the enhanced DFS algorithm with pruning can return the optimal solution to solve the discrete version problem.*

Proof. The enhanced DFS algorithm incorporates three enhancements which improve the performance of the algorithm. The first enhanced approach that we apply is to prune some of the branches as the search graph is traversed based on a greedy heuristic. Traditionally, a DFS algorithm searches a structure by selecting a root node and explores each branch then backtracks. A generalization of the traditional DFS algorithm is considered to be a brute force approach where you search each branch until an optimal solution is returned. We can deem our solution to be a brute force approach with some of the infeasible solution space reduced. Thus, we will prove that our pruning feature in the enhanced DFS algorithm will not eliminate optimal solutions. In each iteration, we check how many new candidate locations $n$ we need to cover the remained fresh points which depends on the sorted order from high to low as tight lower bound. We use the non-equation to check whether we need to prune it or not as following: $n + \text{depth} \geq \text{best}$. If this non-equation is satisfied, we can cut the branch. Specifically, since the number of fresh points is recalculated after each iteration of the search, the cost function is non-decreasing as the search step traverses the graph for a feasible solution. 

\qed
**Algorithm 2 Enhanced-DFS Algorithm**

**Input:** \( L = \emptyset , A = \emptyset , D = \emptyset \)

**Output:** Minimized set of \(|S| = n\), where \( S \) is optimal

*Initialize:* list of \( A , D , L \) and optimal.

1: DFS\((depth)\)
2: \[\text{if } depth \geq best \text{ then} \]
3: \[\text{return} \]
4: \[\text{end if} \]
5: \[\text{if } A == 0 \text{ then} \]
6: \[\text{Update } best; \]
7: \[\text{Record } optimal; \]
8: \[\text{else} \]
9: \[\text{for each } C_L \text{ in } L \text{ do} \]
10: \[\text{Compute all } face \text{ for } C_L \text{ with consideration of obstacles.} \]
11: \[\text{Select } face \text{ with max } F. \]
12: \[\text{Store } C_L \text{ with max } F. \]
13: \[\text{end for} \]
14: \[\text{Sort } L \text{ by descending order of } F; \]
15: \[\text{Store } C^*_L \rightarrow Queue; \]
16: \[\text{while } Queue \neq \emptyset \text{ do} \]
17: \[\text{C^*} \leftarrow \text{Dequeue}; \]
18: \[\text{if } (\text{lowerBound}(C_L(F)^*) + depth \geq optimal) \text{ then} \]
19: \[\text{break}; \]
20: \[\text{end if} \]
21: \[\text{Record } C^*_L \text{ and } F^* \]
22: \[\text{Remove } C^*_L \text{ from } L \text{ and } F^* \text{ from } A; \]
23: \[\text{DFS}(depth + 1) \]
24: \[\text{Add } F^* \text{ back to } A; \]
25: \[\text{end while} \]
26: \[\text{Add all removed } C^*_L \text{ back to } L \]
27: \[\text{end if} \]
28: \[\text{return } optimal \]
Simulations are extensively used to verify the correctness of designs. Our simulation environment is built using both JavaScript and HTML, where the core deployment greedy heuristic and enhanced DFS algorithms are implemented using Java. The evaluation simulator is implemented using JavaScript to visualize and emulate the actual deployment and coverage. The random algorithm serves as a baseline, whose implementation details will be provided in section 6.1. In our evaluation we test three algorithms: random, greedy heuristic and enhanced DFS. The enhanced DFS algorithm is allowed to run up to a time limit of 30 minutes to return the currently found best solution and it can actually return the optimal solution if it finishes within the time limit. We summarize the default setting of our simulations in Table 6.1.

6.1 Baseline Approach

Initially, to tackle the problem we designed a random algorithm, which also serves as a baseline to evaluate our later solutions. In the random algorithm, we randomly choose a location in $L$ and deploy a video sensor at that position. We then adjust the video sensor’s facing direction so that it can cover a maximized number of fresh points (i.e., points that have not been covered by any other deployed video sensor yet) in $A$. After that, we continue to choose another random location and deploy a video sensor there until all the grid points in $A$ have been covered. It is worth noting that during this process, if the video sensor at the randomly chosen location can not cover any fresh point, we will remove the video sensor from this location and choose another random location instead.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D indoor space</td>
<td>$Length \times 60 \times 100$</td>
</tr>
<tr>
<td>A</td>
<td>Same as 3D indoor space</td>
</tr>
<tr>
<td>L</td>
<td>Top half of walls and ceiling</td>
</tr>
<tr>
<td>O</td>
<td>For each $O \in O$, $20 \times 20 \times 30$, one obstacle per 50 unit length</td>
</tr>
<tr>
<td>$g_A$</td>
<td>20</td>
</tr>
<tr>
<td>$g_L$</td>
<td>25</td>
</tr>
<tr>
<td>$g_D$</td>
<td>45</td>
</tr>
<tr>
<td>FOV</td>
<td>50 degrees</td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.7778</td>
</tr>
<tr>
<td>Near Field</td>
<td>1</td>
</tr>
<tr>
<td>Far Field</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 6.1. Simulation parameter settings

6.2 Simulation Results

Fig. 6.1 shows how the number of video sensors change as a function of the length of the 3D indoor space (which can be deemed as the corridors of various length). Also, an actual simulation of the wireless video sensor network deployment for 3D indoor space coverage with consideration of obstacles in the environment, it runs on our JavaScript simulator, where all the grid points within the region that must be fully covered (same as the 3D indoor space in this case) by at least one video sensor in the network is illustrated in Fig. 6.2. Fig. 6.3 shows the performance of each algorithm by the varying the FOV. There is significant reduction in the number of $S$ for both the greedy heuristic and enhanced DFS algorithm. In Fig. 6.4 and Fig. 6.5, the performance for near field and far field variations is explored. The enhanced DFS requires up to 56% less $S$ than the random algorithm and performs better than the greedy heuristic algorithm. The granularity of the grids in $A$, $D$ and $L$ are evaluated in Fig. 6.6-6.8. The performance of the enhanced DFS is stable and continues to reduce the amount of $S$ for optimal coverage in the 3D regions. In Fig. 6.9, we evaluated the different domain variations of candidate location (i.e limit the deployable area to smaller sections of the wall such that 1/5 of upper wall). The x axis dimension for Fig. 6.9 means percentage of walls we can use to deploy video sensors such as, 0.2 means 20 percent of
upper walls. In summary, it is clear to see that both our greedy heuristic algorithm and enhanced DFS algorithm outperform the random algorithm. In particular, compared to the random algorithm, the number of required video sensors can be reduced up to 47% by our greedy heuristic algorithm, and our enhanced DFS algorithm can further reduce the number of video sensors by up to 25% over our greedy heuristic algorithm. We also generated a solution for the second floor of Computer and Information Science department of University of Mississippi showing in Fig. 6.10. The plane figure of second floor of Wire Hall is illustrated in Fig. 11. In this solution, we need 48 video sensors to cover the entire corridor.

![Graph showing number of video sensors as a function of the length of the 3D indoor space](image)

**Figure 6.1.** Number of video sensors as a function of the length of the 3D indoor space
Figure 6.2. Actual simulation of the WVSN deployment for 3D indoor space coverage with obstacles
Figure 6.3. Impact of varying FOVs
Figure 6.4. Impact of varying near fields
Figure 6.5. Impact of varying far fields
Figure 6.6. Granularity of $g_A$ in 3D regions
Figure 6.7. Granularity of $g_D$ for the facing direction sphere
Figure 6.8. Granularity of $g_L$ in 3D regions
Figure 6.9. Impact of varying candidate locations for video sensors
Figure 6.10. Solution for second floor of Computer and Information Science department of University of Mississippi
Figure 6.11. Plane figure of Second floor of Computer and Information Science of University of Mississippi
CHAPTER 7
CONCLUSION AND FUTURE WORK

In this thesis, we studied the problem of 3D indoor space coverage with the consideration of obstacles for video sensor deployment in WVSNs. We first modeled the basic deployment problem of optimizing the placement and facing direction of each wireless video sensor in a continuous space. We then proposed a lattice grid model for a discrete real world applicable environment. We designed two strategies to tackle the additional challenges caused by obstacles which are Divide and Conquer Detection strategy and Accurate Detection strategy. With obstacle-awareness strategies, we can generate more realistic and precise result. We developed both a greedy heuristic and enhanced DFS method to cover all regions whereby we minimized the number of video sensors by careful selecting the location (placement) and direction (angle) of each video sensor. The greedy heuristic algorithm we have proposed can generate reasonably good solution very quickly when compare with the Enhanced DFS, which is further used to improve our enhanced DFS algorithm with other pruning techniques. Extensive simulations driven by the JavaScript emulator showed the enhanced DFS algorithm was able to reduce the number of video sensors required to cover the entire 3D indoor coverage region, which will significantly improve feasibility and outlines the theoretical constraints for 3D indoor space coverage.

We are currently conducting more simulations to further evaluate and improve our solutions. In addition, we will also consider other important issues in WVSNs deployment for 3D indoor space coverage, such as network connectivity, in-network traffic and network lifetime. We also plan to investigate the k-coverage problem for reliability and fault-tolerance. This research can be applied in areas, specifically campus security. Security is a major issue
for college campuses as well as a leading headline in the news. Applying the coverage problem to security concerns as mentioned by Aggarwal (1984) is not novel. However, having the ability to monitor 3D indoor space efficiently by reducing the number of video sensors in a fixed area using optimal placement and directional angle with obstacle-awareness is still an important topic under explored.
BIBLIOGRAPHY


Andersen, T., and S. Tirthapura (2009), Wireless sensor deployment for 3d coverage with constraints, in Networked Sensing Systems (INSS), 2009 Sixth International Conference on, pp. 1–4, IEEE.


Kouakou, M. T., K. Yasumoto, S. Yamamoto, and M. Ito (2012), Cost-efficient sensor deployment in indoor space with obstacles, in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, pp. 1–9, IEEE.


VITA

Zhonghui Wang

Zhonghui Wang is a master student in computer and information science department of University of Mississippi. He got the bachelor degree in Harbin Sport University (Harbin, China).