Optimal 2D-LiDAR-Sensor Coverage of a Room

Noel D'Avis
RheinMain University of Applied Sciences
Wiesbaden, Germany
noel.davis@hs-rm.de

Silvia Faquiri
RheinMain University of Applied Sciences
Wiesbaden, Germany
Fraunhofer IGD
Darmstadt, Germany
silvia.faquiri@hs-rm.de

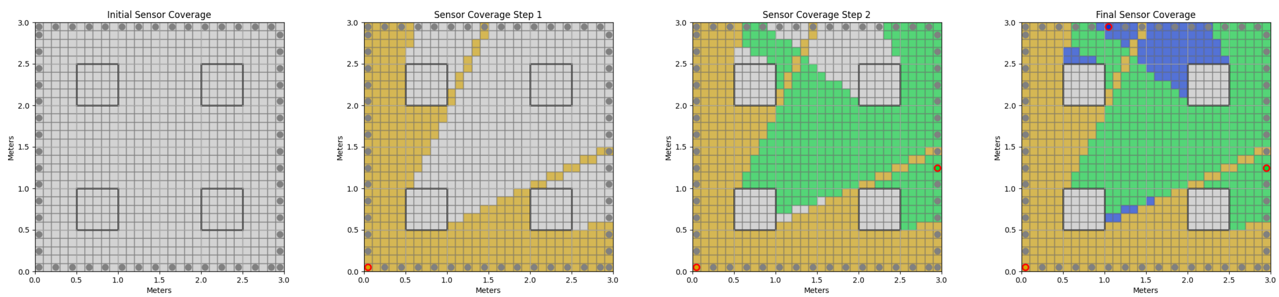

Figure 1: The image shows the steps of the sensor coverage optimization algorithm, where three sensors (yellow, green, blue coverage area) are placed in a square room with four obstacles. The image on the right illustrates the final optimized sensor placement with a minimum number of sensors. Each sensor is strategically positioned to ensure maximum coverage despite obstacles.

Abstract

This work presents a novel concept for achieving optimal coverage of an unspecified room using 2D-LiDAR (Light Detection and Ranging) sensors. The primary goal is to maximize coverage with the fewest possible sensors. We present an algorithm that determines the ideal locations for these sensors, which are all mounted on the floor by the walls. By dividing the room into a grid with adjustable cell sizes (e.g., 10x10 cm), the algorithm marks all grid cells detected by each potential sensor location. This process is repeated for all possible locations. Based on the resulting coverage map, the algorithm calculates the minimum number of required sensors and their optimal positions. An application case for this approach is movement and fall detection using 2D-LiDAR.

CCS Concepts

• **Applied computing → Health care information systems**; **Health informatics**; **Consumer health**; • **General and reference** → *Empirical studies*; *Metrics.*

Keywords

2D-LiDAR, Optimal Coverage, Sensor Placement, Room Monitoring, Grid Mapping, Algorithm Optimization, Movement Detection, Fall Detection, Smart Environments, Indoor Monitoring

ACM Reference Format:
Noel D'Avis and Silvia Faquiri. 2024. Optimal 2D-LiDAR-Sensor Coverage of a Room. In *Proceedings of international Workshop on Sensor-Based Activity Recognition and Artificial Intelligence (iWOAR 2024).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Achieving comprehensive coverage of indoor spaces with minimal sensor usage is crucial for applications such as security, monitoring, and smart environments [1]. The advent of 2D-LiDAR sensors has opened new possibilities for efficient room monitoring due to their precise range-finding capabilities and wide field of view. However, placing these sensors optimally to maximize coverage while minimizing their number remains a significant challenge.

An exemplary application of this concept is in movement and fall detection within indoor environments, as discussed by Bouazizi et al. [1]. Their research highlights the use of multiple 2D-LiDAR sensors for activity detection, demonstrating the practical relevance of optimal sensor placement for efficient monitoring. In scenarios such as elderly care or security monitoring, accurately detecting movement is critical.

iWOAR 2024, September 26–27, 2024, Potsdam

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

The proposed algorithm can ensure comprehensive coverage of rooms and common areas with minimal sensors, enabling quick detection and response to any incidents.

2 Related Work

Optimal sensor placement aims to maximize coverage with a minimal number of sensors, which is crucial for cost-effective and efficient monitoring. Various approaches have been proposed to address this challenge. For instance, Dhillon and Chakrabarty developed a framework for sensor placement in wireless sensor networks, focusing on coverage and connectivity issues [3]. Their approach utilizes a grid-based method to ensure that the sensors' coverage areas overlap sufficiently to detect all events of interest.

In the field of robotics, Gonzalez-Banos and Latombe (2001) introduced an art gallery approach to place sensors in a polygonal environment [4]. Their algorithm ensures that the entire area is covered by a minimal number of sensors, leveraging the visibility properties of polygons. This work has been foundational in developing algorithms for sensor placement in indoor environments.

Recent advancements in LiDAR technology have enabled more precise and flexible sensor placement strategies. Bouazizi et al. (2024) explored the use of multiple 2D-LiDAR sensors for activity detection in indoor environments, highlighting the importance of optimal sensor placement for efficient monitoring [1]. Their research demonstrates that strategically placed LiDAR sensors can provide comprehensive coverage and accurate activity detection with fewer sensors.

In this work, we present a conceptual approach to address this challenge by developing an algorithm that determines the ideal locations for 2D-LiDAR sensors placed on the floor of a room by the walls. The algorithm ensures that the entire room is covered with the fewest sensors possible, thus optimizing both cost and performance. The proposed approach leverages a centralized software system that records room dimensions and sensor positions, enabling efficient and effective room monitoring.

3 Sensor Placement Algorithm

The process of achieving optimal coverage by placing 2D-LiDAR sensors involves several steps: First, the dimensions of the room are entered into the software (Listing 1). This system maintains a detailed map of the room and records all possible sensor locations. The sensors are then placed at the various potential positions along the room's walls. These positions are input into the software, and each sensor is initially marked as "non-essential". These virtual sensors perform a series of measurements to detect walls or obstacles at various angles.

When a sensor detects an obstacle at a specific angle, all cells between the sensor and the obstacle at that angle are marked as covered. This data is fed back into the software, where each cell in the grid maintains an array of sensors that cover it. Additionally, each sensor is assigned a value based on the number of cells it covers. The algorithm evaluates all potential sensor positions to create a comprehensive coverage map. This involves iterating through each sensor and updating the coverage information for each cell in the grid.

Listing 1: Optimal 2D-LiDAR Sensor Placement Algorithm Pseudocode

```
# Input: Room dimensions, grid cell size,
    sensor range
# Output: Optimal sensor locations

# Initialize grid based on room dimensions
    and grid cell size
grid = initialize_grid(room_dimensions,
    grid_cell_size)
coverage_map = initialize_coverage_map(grid)

# Evaluate potential sensor positions
for each possible sensor location in room:
    for angle in sensor_angles:
        cells_covered = get_covered_cells(
            sensor_location, angle,
            sensor_range)
        update_coverage_map(coverage_map,
            sensor_location, cells_covered)

# Determine essential sensors
essential_sensors = []
for cell in grid:
    if is_covered_by_single_sensor(cell,
       coverage_map):
        sensor = get_covering_sensor(cell,
            coverage_map)
        mark_as_essential(sensor,
            essential_sensors)
        mark_cells_as_completed(cell,
            coverage_map)

# Optimize sensor placement
while not all_cells_covered(grid,
    coverage_map):
    sensor = get_sensor_with_most_coverage(
        coverage_map)
    mark_as_essential(sensor,
        essential_sensors)
    update_coverage_map_after_selection(
        sensor, coverage_map)

# Return optimal sensor locations
return essential_sensors
```

Once the initial coverage map is created, the algorithm determines the necessary sensor locations through an optimization process. For cells covered by only one sensor, that sensor is marked as "essential" and the cells are marked as "completed." Using a "First Fit Decreasing" strategy, the sensor covering the most cells is then marked as "essential." After each step, the number of cells covered by each sensor is adjusted by subtracting cells already covered by other sensors. This process continues iteratively until all cells are covered.

The final outcome of the algorithm is a set of sensors, each with a specific optimal position. These positions ensure that the entire room is covered with the fewest number of sensors, optimizing both cost and performance. We refer to the pseudocode in Listing 1 for a detailed algorithmic representation of the optimal LiDAR sensor placement process.

3.1 Implementation Details

The implementation of the proposed algorithm involves several key steps, each critical to achieving optimal sensor placement.

Grid Division. The room is divided into a grid based on specified cell sizes. This grid forms the basis for all subsequent calculations and evaluations. Each cell in the grid represents a small section of the room, and the size of these cells can be adjusted based on the specific requirements of the environment and the capabilities of the sensors.

Sensor Range Calculation. For each potential sensor location, the algorithm calculates the range of cells that the sensor can cover. This calculation takes into account the sensor's maximum range and its angular field of view.

Coverage Map Update. As each sensor performs measurements and detects obstacles, the coverage map is updated. The software records which cells are covered by which sensors, and updates the array of sensors covering each cell. This ensures that the coverage map is accurate and up-to-date.

Optimization Algorithm. The optimization step is crucial for determining the minimal set of sensors needed to cover all cells. Techniques such as integer linear programming (ILP) or greedy algorithms can be employed to solve this problem. The "First Fit Decreasing" strategy is particularly effective, as it prioritizes sensors that cover the most cells, ensuring efficient coverage.

4 Preliminary Algorithm Validation

To validate the proposed algorithm, simulations were conducted in various obstacle configurations. The simulations involved different grid cell sizes and numbers of obstacles. The results showed that the algorithm consistently identified the optimal sensor placements, achieving full coverage with a minimal number of sensors.

The simulations were performed in a virtual environment where grid cell sizes, and number of obstacles were varied (Table 1). The algorithm was tested for its ability to cover the entire room with the fewest sensors. Various scenarios were simulated to account for different room shapes and obstacle placements.

The algorithm's performance was evaluated through a series of simulations conducted in a square environment with varying numbers of obstacles: specifically, 1, 4, or 7. Determining sensor placement in an unobstructed square space is straightforward. However, the introduction of obstacles significantly increases the complexity of the problem.

Table 1: Simulation setups of square space with different cell grid sizes and variying number of randomly placed obstacles

Room Size	Grid Cell Size	Number of Obstacles
3x3 m	5x5 cm	1
3x3 m	10x10 cm	1
3x3 m	25x25 cm	1
3x3 m	33x33 cm	1
3x3 m	5x5 cm	4
3x3 m	10x10 cm	4
3x3 m	25x25 cm	4
3x3 m	5x5 cm	7
3x3 m	10x10 cm	7
3x3 m	25x25 cm	7

Figure 2a, shows the initial plot of a 3 m x 3 m space with a single obstacle, representing the second simulation configuration from Table 1. Potential sensor locations are shown along the walls placed 20 cm apart, a total of 24 sensors in this case. The space is divided into 10 cm grid cells. The algorithm checks if there are cells that are only covered by one sensor. Due to the high resolution in this simple simulation configuration, no such cells exist. In the next step (Figure 2b), the sensor with the highest number of covered cells is marked as "necessary". In this case it is the sensor at position (0, 0), the red dot in left corner. All covered cells are marked as covered, colored dark blue with 240 grid cells remaining uncovered. This step is repeated identifying the sensor that covers the second highest number of cells, in this case at position (30, 20). This area is colored light blue. The process is repeated until all cells are covered. In this illustratory example, the algorithm terminates after 2 steps. The plot of the final step, shown in Figure 2c, displays the two sensors, along with the covered cells.

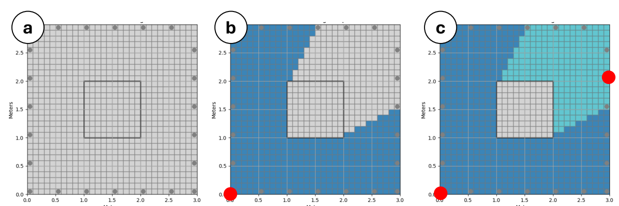

Figure 2: Intermediary algorithm results (b), where 2 sensors are enough (c) to cover the area around the obstacle in the middle (a)

The algorithm was also tested in more complex environments with four and seven obstacles, with results shown in Figure 1 respectively Figure 3. For the 4-obstacles environment three sensors are enough to cover the area. For the 7-obstacles environment five sensors need to be specifically placed.

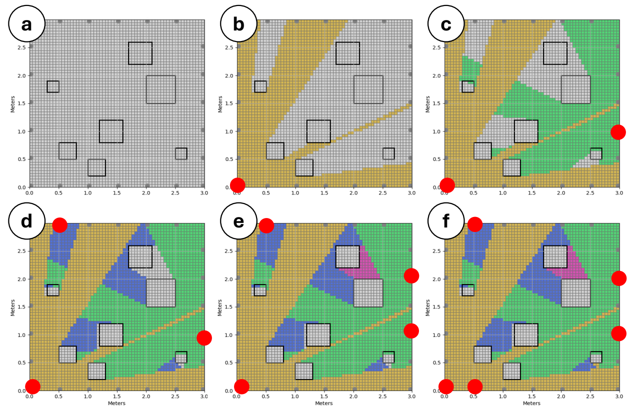

Figure 3: Intermediary algorithm results (b-e) for a 7-obstacle environment (a), where five sensors are needed to cover all the areas between the randomly placed obstacles (f)

5 Discussion

The results of the simulation cases indicate that the algorithm consistently produces favorable outcomes, achieving the minimal number of necessary sensors in all tested scenarios. These were validated against manual calculations and alternative optimization techniques. The "First Fit Decreasing" strategy, combined with efficient data structures, ensures that the algorithm runs efficiently even for large rooms with complex layouts.

A noteworthy aspect of the algorithm is its handling of scenarios where multiple sensor locations provide identical coverage. In such instances, rather than selecting an optimal angle, the algorithm employs a randomized selection process. This approach maintains the algorithm's flexibility and adaptability, even when faced with multiple equally optimal solutions.

The impact of variations in potential sensor locations and grid sizes on the results appears to be marginal. However, these variations warrant more thorough evaluation in future research to ensure a comprehensive understanding and robustness of the algorithm.

While the algorithm performs well in simulations, there are limitations to consider. The accuracy of the coverage map depends on the precision of the sensors and the resolution of the grid. Also, in real-world applications, factors such as sensor malfunctions or environmental changes could affect the performance of the system. Futhermore, there might be areas along the wall, where sensor placement is not allowed or not possible and thus scenarios might arise where no full sensor coverage is achievable. Thus, strategies might be needed to tackle these situations based on context information such as path tracking and fall probability.

5.1 Complexity

In this section, we analyze the complexity of our algorithm with respect to both time and space.

Firstly, we consider the input parameters: the room size (L), which denotes the side length of the square room; the sensor spacing (S), representing the distance between sensors positioned along the walls; and the grid size (G), the dimension of the cells within the grid, for example, 10x10 cm.

For the time complexity, we evaluate three main components. The initialization of the room map and sensors has a complexity of $O\left(\left(\frac{L}{G}\right)^2\right)$. The calculation of line-of-sight and coverage per sensor has a complexity of $O\left(\frac{L^3}{S\cdot G^3}\right)$. The optimization process for selecting the minimal number of sensors has a complexity of $O\left(\frac{L^3}{S\cdot G^2}\right)$. The dominant term in these calculations is $O\left(\frac{L^3}{S\cdot G^2}\right)$, which represents the overall time complexity of the algorithm.

For the space complexity, we again evaluate two main components. The space required for the room map is $O\left(\left(\frac{L}{G}\right)^2\right)$, while the space needed for sensor coverage data is $O\left(\frac{L^3}{S\cdot G^2}\right)$. Therefore, the overall space complexity is $O\left(\frac{L^3}{S\cdot G^2}\right)$.

6 Conclusion and Future Work

This work proposes an algorithm for optimizing the placement of 2D-LiDAR sensors to achieve maximal room coverage with minimal sensors. By leveraging adjustable grid cell sizes, sensor range and angle span, as well as obstacle mapping, the algorithm enables effective monitoring of indoor spaces. The potential applications in various smart environment scenarios, such as movement and fall detection, highlight the practical significance of this research.

The proposed algorithm, validated through simulations, demonstrates a reliable, efficient and flexible way of sensor placement. The algorithm improves the feasibility of 2D LiDAR-based systems and ensures full sensor coverage in real applications.

Further research will test the algorithm with various room layouts and different numbers of potential sensor locations. This will ensure that the algorithm performs optimally across a range of scenarios and configurations. As a next step, the algorithm will be tested using real sensor data rather than simulations, which will provide a more accurate evaluation of its practical applicability and performance.

Furthermore, due to the increasing relevance of 3D-LiDAR sensors [2, 5, 6], exploring the placement of stationary 3D-LiDAR sensors will also be investigated to improve coverage and increase efficiency in complex environments.

References

[1] Mondher Bouazizi, Alejandro Lorite Mora, Kevin Feghoul, and Tomoaki Ohtsuki. 2024. Activity Detection in Indoor Environments Using Multiple 2D Lidars. *Sensors* 24, 626 (2024), 1–27. https://doi.org/10.3390/s24020626 Academic Editor: Natividad Duro Carralero.

[2] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. 2016. Motion-based detection and tracking in 3D LiDAR scans. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Stockholm, 4508–4513. https://doi.org/10.1109/ICRA.2016.7487649

[3] S.S. Dhillon and K. Chakrabarty. 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. *IEEE Wireless Communications and Networking* 3 (2003), 1609–1614.

[4] H.H. Gonzalez-Banos and J.-C. Latombe. 2001. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry*. ACM, Medford, MA, USA, 232–240.

[5] Juan Gómez, Olivier Aycard, and Junaid Baber. 2023. Efficient Detection and Tracking of Human Using 3D LiDAR Sensor. *Sensors* 23, 10 (2023). https://doi.org/10.3390/s23104720

[6] Abhishek Thakur and P Rajalakshmi. 2024. L3D-OTVE: LiDAR-Based 3-D Object Tracking and Velocity Estimation Using LiDAR Odometry. *IEEE Sensors Letters* 8, 7 (2024), 1–4. https://doi.org/10.1109/LSENS.2024.3416411