

# Optimal Placement of Roadside Infrastructure Sensors towards Safer Autonomous Vehicle Deployments

Roshan Vijay<sup>1</sup>, Jim Cherian<sup>1</sup>, Rachid Riah<sup>1</sup>, Niels de Boer<sup>1</sup> and Apratim Choudhury<sup>2</sup>

**Abstract**— Vehicles with driving automation are increasingly being developed for deployment across the world. However, the onboard sensing and perception capabilities of such automated or autonomous vehicles (AV) may not be sufficient to ensure safety under all scenarios and contexts. Infrastructure-augmented environment perception using roadside infrastructure sensors can be considered as an effective solution, at least for selected regions of interest such as urban road intersections or curved roads that present occlusions to the AV. However, they incur significant costs for procurement, installation and maintenance. Therefore these sensors must be placed strategically and optimally to yield maximum benefits in terms of the overall safety of road users. In this paper, we propose a novel methodology towards obtaining an optimal placement of V2X (Vehicle-to-everything) infrastructure sensors, which is particularly attractive to urban AV deployments, with various considerations including costs, coverage and redundancy. We combine the latest advances made in raycasting and linear optimization literature to deliver a tool for urban city planners, traffic analysis and AV deployment operators. Through experimental evaluation in representative environments, we prove the benefits and practicality of our approach.

## I. INTRODUCTION

### A. Motivation

Development of autonomous vehicles (AV) has received global attention in the recent years. An AV consists of a vehicle equipped with an Automated Driving System (ADS) [1]. The ADS utilizes various on-board sensors, data processing and compute capabilities to sense and perceive the state of the world around the AV, to plan and to act, in order to *safely* navigate to a given destination.

The sensing and perception capabilities of an AV by means of on-board sensors alone, may be limited and insufficient for urban environments, due to occlusions or limited range, even when their placement on the AV is optimal [2]. In contrast, combining the data from an AV's on-board sensors with that received from infrastructure-mounted sensors using Vehicle-to-Infrastructure (V2I) communication is a way to enhance safety and performance. Additional benefits of roadside infrastructure

sensors include the possibility that the sensed data can be used to improve traffic coordination and flow, by integrating it with traffic light controller(s).

To accelerate safe AV deployments, infrastructure sensors need to be used in complex urban environments instead of only some controlled testing/trial facilities. This may impose tighter requirements such as complete coverage of intersections with multiple buildings and obstacles in highly built-up areas. Thus, the optimal placement of infrastructure sensors will be a critical factor in enhancing the real-world operational safety of the AV.

### B. Problem Statement

The optimal sensor placement problem (OSP) has been extensively researched for surveillance purposes. In these studies the coverage estimation or the area visible to each sensor is approximated using very simple shapes. For example, in [3], the coverage estimation is approximated using a trapezoid shape, and in [4], [5], the area covered by the visibility model of the sensor is described using straightforward relationships. Current studies approach the problem in a theoretical manner, without real-world representations of actual physical sensors or environments.

In order to address these weaknesses, in this paper, we propose a raycasting-based sensor model to approximate accurate line-of-sight coverage. This results in optimal placement even while considering worst case scenarios with multiple 3D obstacles. To solve the optimization problem, we adopt a regularized Binary Integer Programming (BIP) method. We note that BIP is widely used for such discrete sensor placement optimization problems [3].

### C. Contributions

In this paper, we investigate the problem of optimal placement of roadside infrastructure sensors in realistic road networks and make the following contributions:

- 1) A generic Line-of-Sight (LoS) sensor model designed for raycasting and defined using parameters such as vertical and horizontal field of view and range.
- 2) A visibility pre-processing methodology that helps to estimate sensor coverage of discrete target points defined within an area of interest.
- 3) A framework for optimizing sensor placement to achieve maximum coverage of the given target area while minimizing the number of sensors required.
- 4) An end-to-end empirical evaluation of our proposed framework under representative environments

This work is supported by Siemens Mobility Pte. Ltd.

<sup>1</sup>Authors are with Centre of Excellence for Testing & Research of AVs - NTU (CETRAN), Nanyang Technological University, CleanTech One, 637141, Singapore {rvijay,jcherian,rachid.riah,niels.deboer}@ntu.edu.sg

<sup>2</sup>Author is with Siemens Mobility Pte. Ltd., Singapore apratim.choudhury@siemens.com

The rest of this document is structured as follows: in Section II we briefly review the state of the art on sensor placement, coverage calculation and optimization methods. Section III introduces the system architecture of the proposed approach. In Section IV we describe the theoretical structure of our proposed approach. In Section V, we empirically evaluate our proposed methodology and demonstrate its practical usefulness by means of a reference implementation and several input configurations. We briefly review the potential scope for further improvements and conclude in Section VI.

## II. LITERATURE REVIEW

The optimal sensors placement problem (OSP) in an outdoor environment is defined similar to the art gallery problem (AGP). AGP is a well-studied visibility problem in computational geometry. It defines the real-world problem of guarding an art gallery with a minimum number of guards who can together observe the whole gallery [6], [7]. For outdoor environments, this problem would enquire on how many sensors are necessary to observe an entire Region of Interest (RoI) while considering the presence of obstacles and occlusions. Thus, solving the OSP is similar to solving the AGP [8], [3]. Solving OSP in outdoor environments has received much attention recently, especially to accelerate AV deployments by improving the effective sensor coverage area for a given road network. Several methods have been developed and used in literature to find efficient algorithms to estimate the optimal solution, as the problem is NP-hard [9].

Research efforts in wireless sensor networks (WSNs) have also studied how to model, simulate and maximize the coverage of a WSN. In [10], Argany et al. proposes an environment framework using a geographic CityGML model of a region which is then converted to either a raster or vector format for estimating sensor coverage. They estimate sensor coverage by quantifying the number of tiles on the raster map which can be ‘seen’ by each sensor. In this approach, the granularity of tile dimensions exerts a strong influence on the performance.

[11] adopts a similar approach by modeling each sensor as a directional probabilistic model, by handling obstacle and topographical occlusions through line-of-sight estimation, and by optimizing the placement of sensors using a gradient descent approach. [12] involves a grid cell based environment classification approach whereby certain cells are considered to be obstacles or occlusions based on data from a 2D map environment. Using this, 2D sensors are modelled and coverage is estimated for optimizing their placement using a genetic algorithm. [13] proposes a WSN coverage estimation algorithm based on landmark points, where they act as sampling points for an irregularly shaped RoI. Other studies have also used a similar landmark or target point based approach [14], [3]. [3] applies this approach in two phases, first to find the minimum number of sensors using a low resolution grid of target points, and then to maximize coverage by utilizing

a higher resolution grid of target points. One common theme among all these studies is that the target point approach seems to be well suited to irregularly shaped RoIs where other regular geometrical coverage estimation methods cannot be utilized.

In terms of optimization algorithms, two different types of OSP formulations have been studied: the MIN and the FIX problems [15]. The MIN problem aims to minimize the number of sensors needed to cover the entire RoI. The FIX problem aims to maximize the covered area for a *fixed* number of sensors. In this paper, we adopt the MIN approach. The challenges in solving these two problems are well understood and many approximate solutions are proposed in the literature [15].

Most studies adopt Binary Integer Programming (BIP) [4], [16] as a preferred tool to solve OSP. The advantage of BIP is that the solution of the optimization problem converges to a global optimum and can be used to solve both the FIX and MIN problems depending on formulation. Depending on the coverage estimation methods and the real-world conditions, the optimization algorithms proposed may vary. The commonly used approximation optimization approaches to solve these include genetic algorithms [17], greedy algorithm [16], particle swarm optimization [18], [19], and direct search (DS) method.

In [20] a raycasting-based approach for area coverage estimation is suggested. They use an optimization algorithm applying the Direct Search (DS) method to solve the FIX problem. In our paper, we propose to apply BIP optimization along with raycasting for area coverage estimation to find the optimal number of sensors with their exact positions. Our motivation to use BIP instead of DS is primarily because DS is more well suited to solve the FIX problem, where maximizing the coverage area is less of a priority than fixing the total number of sensors.

BIP offers the following key advantages: 1) The MIN problem can be solved, 2) large number of sensors can be considered. On the contrary, the time to converge to an optimal solution can be higher than other methods. Fortunately, this is not a concern for us as our proposed framework is intended to work offline and offsite without any need for instantaneous results.

Our methodology can be of particular interest to practitioners due to its relative simplicity and direct applicability to real-world urban environments and road networks. It requires only a virtual 3D model (‘digital twin’) of the RoI where the sensors are to be deployed, and it can compute an effective solution while taking real-world dimensions, obstacles, sensor parameters and other constraints into account.

## III. SYSTEM ARCHITECTURE

The overall system architecture of our proposed approach is illustrated in Fig. 1. The inputs are the environment parameters and candidate sensor parameters. The former may include a 3D environment model, road geometry parameters and information about obstacles (if

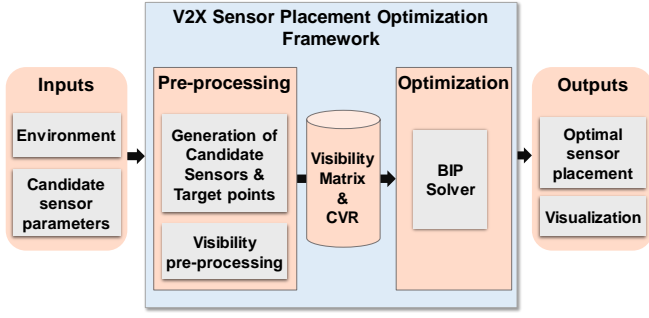


Fig. 1: System architecture for the proposed framework

any). The latter forms a superset of ‘candidate’ sensors from which the optimization algorithm can attain the final set of optimal sensor positions. Given these inputs, the framework prepares optimizer inputs required by the BIP solver by generating the candidate sensors, estimating their coverage represented as a visibility matrix produced by the visibility pre-processing step, and calculating the maximum feasible coverage ( $CVR$ ) with an initial superset of candidate sensors. Finally, the solver is invoked and the optimal outputs are generated. The final output is an optimal set of sensor positions along with its visualization.

#### IV. METHODOLOGY

##### A. Environment model

The environment model consists of:

- Road network
- Road features i.e. lamp posts or sign posts
- Obstacles such as buildings and static vehicles

##### B. Sensor model

In order to estimate sensor coverage, we use raycasting [21], a well known computer graphics technique. It involves sending out virtual rays of light from a diffuse or point source to calculate obstacles in its line-of-sight. In our case, the sensor is modelled as a point source with multiple rays emanating from it. The maximum raycasting distance can be specified as the sensor range.

The horizontal coordinate system is used to determine the ray direction vector. This is illustrated in Fig. 2.

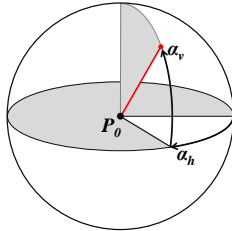


Fig. 2: Horizontal coordinate system for ray coordinates

For a ray emanating from an origin  $P_0$ , the direction vector  $\theta$  of the ray is a function of its horizontal angle  $\alpha_h$  and its vertical angle  $\alpha_v$ :

$$\theta = [\sin(\alpha_h)\cos(\alpha_v), \cos(\alpha_h)\cos(\alpha_v), \sin(\alpha_v)] \quad (1)$$

##### C. Visibility Pre-Processing

Once the sensor cast points have been determined, the visibility pre-processing step is necessary to solve the optimization problem. During visibility pre-processing, the list of covered target points is used to construct a binary matrix called the visibility matrix  $V$ . Its rows and columns represent the candidate sensor locations and covered target points, respectively. A binary value is assigned to each element of  $V$  based on if the target point at a particular position is visible to a particular sensor. If the sensor cast point lies within a certain distance to the target point, it is considered visible.

Therefore, an individual element  $v_{ik}$  of the visibility matrix  $V$  is defined as:

$$v_{ik} = \begin{cases} 1, & \text{if the target point at position } k \\ & \text{is covered by the sensor at position } i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

From the visibility matrix  $V$ , the maximum feasible coverage  $CVR$  with all the candidate sensors present is calculated as:

$$CVR = \frac{\text{Number of covered target points}}{\text{Total number of target points}} \quad (3)$$

Hence,  $CVR \in [0 \ 1]$

The solution of the optimization problem will be a minimum number of sensor positions achieving this  $CVR$  value. In an ideal case, without occlusions, the maximum feasible  $CVR$  should be 1.

If  $CVR < 1$  with no obstacles, the number and/or placement of the candidate sensors need to be redefined in order to effectively cover the target region.

##### D. Optimization Framework

###### 1) Assumptions

The optimization problem is addressed in a deterministic manner where the solution comprising of the minimum number of sensor positions is selected from a set of discrete candidate sensor positions. To ensure that the problem can be applied to real-world scenarios with specific and finite sensor installation locations, a non-deterministic strategy based on randomly selecting candidate sensor positions was not considered.

We assume that the sensors are homogeneous and thus have identical properties such as field-of-view and range and are spaced uniformly within the purview of the permitted candidate sensor locations. We also assume that the target points are defined within an RoI but may not cover the entire RoI and that they are equally spaced and homogenous with respect to the sensor coverage criteria, i.e. all target points have the same criteria by which they are considered to be covered by or are ‘visible’ to a sensor.

The optimization framework is developed and implemented with the following assumptions:

- 1) The RoI consists of the 3D environment, road network and grid of target points located within

the road network (see Fig. 3). The target points are discrete points in a uniform 2D distribution.

- 2) Raycasting is performed across the entire RoI.
- 3) A target point is considered visible to the sensor if it is observable from the sensor based on the criteria mentioned in Section IV-C.
- 4) Sensors cannot be placed within the road network. They are defined outside the road boundaries.
- 5) All sensors are placed at a specified uniform height  $h$  above the ground plane.

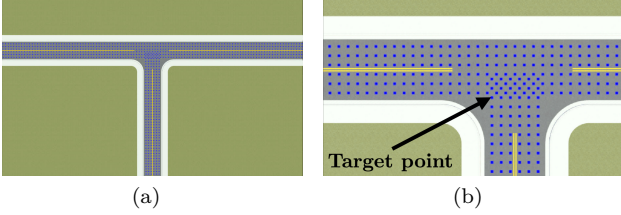


Fig. 3: (a) Illustration of target points (blue dots) on road network, (b) A closer view

## 2) The optimization algorithm

The optimization problem is formulated as a BIP problem with binary integer values assigned to the decision variables and is expressed as a linear programming problem. The detailed formulation is given below:

- 1) The variables  $N_S$  and  $N_T$  are defined as:
  - a)  $N_S$  = number of sensor candidate positions
  - b)  $N_T$  = number of target points

- 2) The decision variables  $s_i$  and  $c_k$  are defined as:

$$s_i = \begin{cases} 1, & \text{if there exists a sensor at position } i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$c_k = \begin{cases} 1, & \text{if the target grid point at position } k \\ & \text{is covered by atleast one sensor} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

- 3) The objective function is defined as:

$$\min \left\{ \sum_{i=1}^{N_S} s_i + \lambda \sum_{i=1}^{N_S} D_i \cdot s_i \right\} \quad (6)$$

Where the first term is the main objective function for minimization of sensors and the second is a regularization term. In the regularization term,  $D_i$  is a sum of overlap decisions across  $s_i$ , and  $\lambda$  is a calibration parameter. This is introduced to reduce overfitting-like effects such as the selection of very closely spaced sensor positions.

- 4)  $D_i$  is given by:

$$D_i = \sum_{j=1}^{N_S} O_{ij} \quad (7)$$

$$O_{ij} = \begin{cases} 1, & \text{if } \|P_j - P_i\| \leq L \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$L$  denotes the desired minimum distance between two candidate sensor positions  $P_i$  and  $P_j$  for avoiding overlap.

- 5) The first constraint ensures that if the target point  $c_k$  is covered then there exists at least one sensor at position  $s_i$  to which this target point is visible:

$$\sum_{i=1}^{N_S} v_{ik} \cdot s_i \geq 0 \quad \text{for } \forall k \in (1, \dots, N_T) \quad (9)$$

where  $v_{ik}$  is defined in Eq. 2.

- 6) The second constraint ensures that if the target point  $c_k$  is covered, then there exists a maximum of  $N_S$  sensors to which this target point is visible:

$$\sum_{i=1}^{N_S} v_{ik} \cdot s_i \leq N_S \cdot c_k \quad \text{for } \forall k \in (1, \dots, N_T) \quad (10)$$

These constraints are necessary in order to get the value of  $c_k$  in Eq. 11.

- 7) The third constraint, the coverage constraint, guarantees that at least  $N_T \cdot CVR$  target grid points will be covered:

$$\sum_{k=1}^{N_T} c_k \geq N_T \cdot CVR \quad (11)$$

We note that visibility pre-processing as described in Section IV-C helps to make the constraints 9 and 10 linear such that the BIP problem can be efficiently solved.

## V. EXPERIMENTAL EVALUATION

### A. Experimental Setup

An experimental setup for the sensor placement optimization framework was implemented using Python3 scripts. The environment models for raycasting were created using MathWorks RoadRunner [22] and were imported in the GLTF format [23]. Trimesh Python library [24] was used to process the GLTF files and perform raycasting. The Python API of lpsolve [25], a Mixed Integer Linear Programming (MILP) Solver, was used to solve the BIP problem (refer Section IV-D.2). Finally, the optimal results were visualized using the PyVista library [26].

### B. Test inputs

Six configurations were generated from three candidate sensor arrangement variants, either with or without obstacles. For each configuration, multiple experiments were performed while varying parameters such as sensor height and maximum attainable CVR. They are summarized in Table I. The sensor was modelled with parameters of a commonly used LIDAR sensor [27], with the following values:

- Vertical field of view:  $-17^\circ$  to  $+3^\circ$  in  $1^\circ$  increments

Config	Candidate spacing	Candidate sensor count ( $N_S$ )	Obstacles
A1	4x4 m	79	0
A2	4x4 m	79	4
B1	5x5 m	63	0
B2	5x5 m	63	4
C1	6x6 m	51	0
C2	6x6 m	51	4

TABLE I: Test inputs and parameters

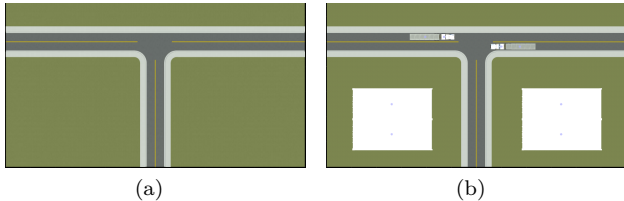


Fig. 4: Road geometry environment model: (a) without obstacles (b) with multiple obstacles

- Horizontal field of view:  $0^\circ$  to  $360^\circ$  in  $1^\circ$  increments
- Specified range: 100m
- Actual range at 2.4m height:  $\approx 67$ m

Since the sensor model is generic, one may use this to configure other types of LoS sensors as well.

### C. Environment setup

The evaluation was carried out with two environment models shown in Fig. 4a and 4b. Both represent a simple T junction with two roads; one with only the 2D road geometry and the other with two sets of oversized containers and two trucks with semi-trailers added as occlusions. The target points were configured such that they cover only the asphalt section of the road. The target points were configured with the following parameters:

- Target point spacing: 1x1m
- Target point radius: 1m
- Total number of target points: 1008

### D. List of experiments

#### 1) Visibility pre-processing performance

We assess the performance of the visibility pre-processing algorithm by varying parameters relating to the candidate sensor configuration and environment, and evaluating the coverage estimation by calculating the percentage mean coverage of the target points. This refers to the percentage mean count of sensors that can ‘see’ each target point within the RoI. The percentage coverage  $C_i$  at the  $i$ th target point, is calculated as follows:

$$C_i = \frac{N_{Si}}{N_S} \times 100 \quad (12)$$

where  $N_{Si}$  is the count of sensors visible at the target point  $i$  and  $N_S$  is the count of the total number of candidate sensors. We compute the mean of  $C_i$  across all the target points to obtain the percentage mean coverage of the

selected sensor and configuration. Similarly we employ another metric, namely, the percentage median coverage, to estimate the coverage performance of the selected sensors after optimization.

We calculate the percentage mean coverage by:

- Varying sensor height
- Varying sensor spacing
- Varying the presence of obstacles

Figs. 5a, 5b and 5c show the visibility pre-processing performance for the candidate sensor configurations A, B and C respectively with corresponding spacings 4x4m, 5x5m and 6x6m. Separate cases with and without obstacles (A1,A2), (B1,B2) and (C1,C2) are also evaluated. It is evident that the presence of obstacles in the environment greatly reduces the percentage mean coverage performance.

#### 2) Optimization performance

To evaluate our proposed optimization approach, we calculate the percentage mean coverage with differing sensor heights for the A2, B2 and C2 configurations as specified in Table I.

The final optimal sensor configurations for these candidate configurations are visualized in 6a, 6b and 6c respectively. The selected sensor count is significantly lower than the candidate sensor positions. We can also see that closer spaced candidate sensors lead to fewer selected optimal sensors. In Fig. 6b, two sensors are found to overlap near the left-side obstacle. This is because our cost function factors the overlapping of sensors as a regularization parameter and not as a hard constraint. We use the percentage mean coverage to compare the initial set of candidate sensors and the final set of selected ‘optimal’ sensors, as shown in Figs. 7a, 7b and 7c for the configurations A2, B2 and C2 respectively. In order to analyze the optimization performance in a worst case scenario, we choose only the environment model with obstacles. Through this performance comparison, one can discern that that after optimization, the number of redundant sensors covering each target point has greatly reduced; this is because only a fraction of the large set of candidate sensors are selected while maintaining the same *CVR* of 100%. However, we must also consider that even with 100% coverage, the redundancy at a particular target point may be significantly reduced as the sensor count has been minimized. Fig. 8a shows the percentage median coverage (an indicator of redundancy in terms of sensor coverage) of different configurations after optimization. Clearly, the spacing of the candidate sensors has a direct correlation to the coverage per target point. Closer spaced candidate sensors result in better coverage of each target point after optimization, as the solver has more candidates at its disposal to evaluate and find an optimal subset.

## VI. CONCLUSION

In this paper, we have proposed a novel raycasting-based framework for estimating sensor coverage and



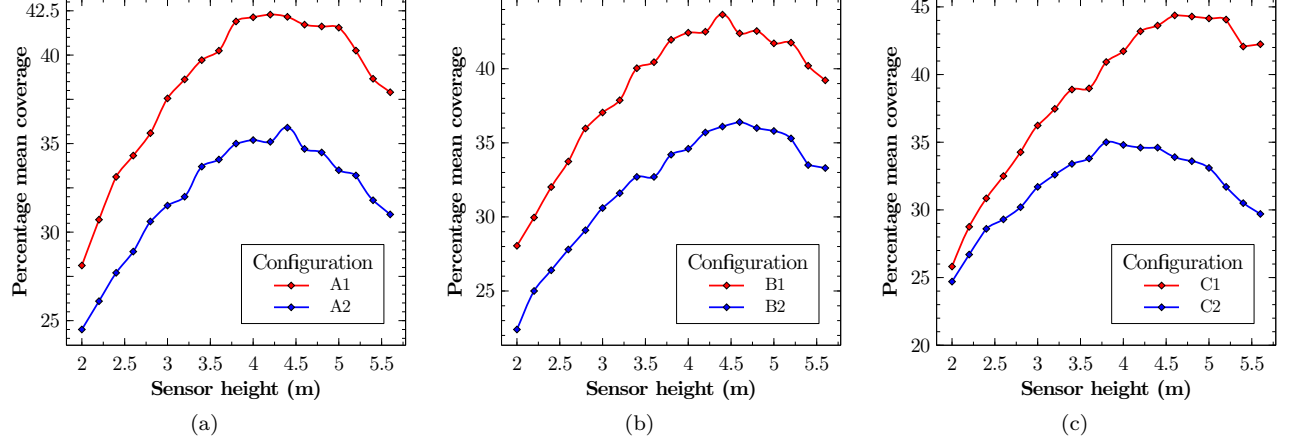


Fig. 5: Visibility pre-processing coverage performance for sensor configurations: (a) A1, A2 (b) B1, B2 and (c) C1, C2, under different sensor heights

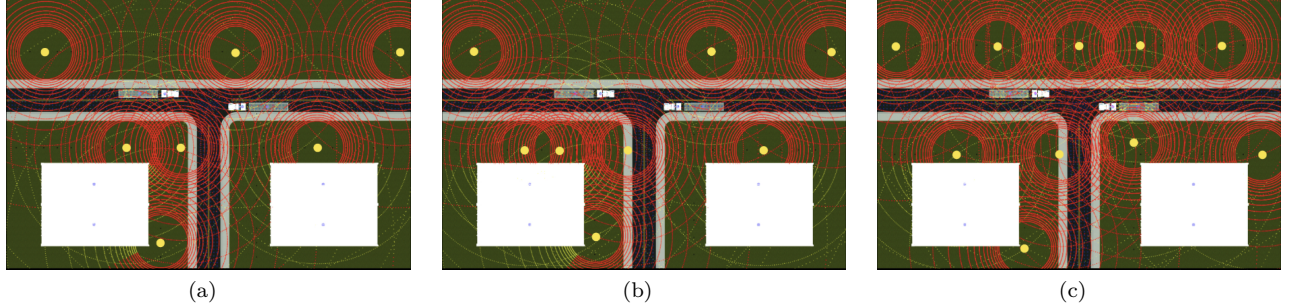


Fig. 6: Visualization of optimal results for candidate sensor configurations: (a) A2, (b) B2 and (c) C2

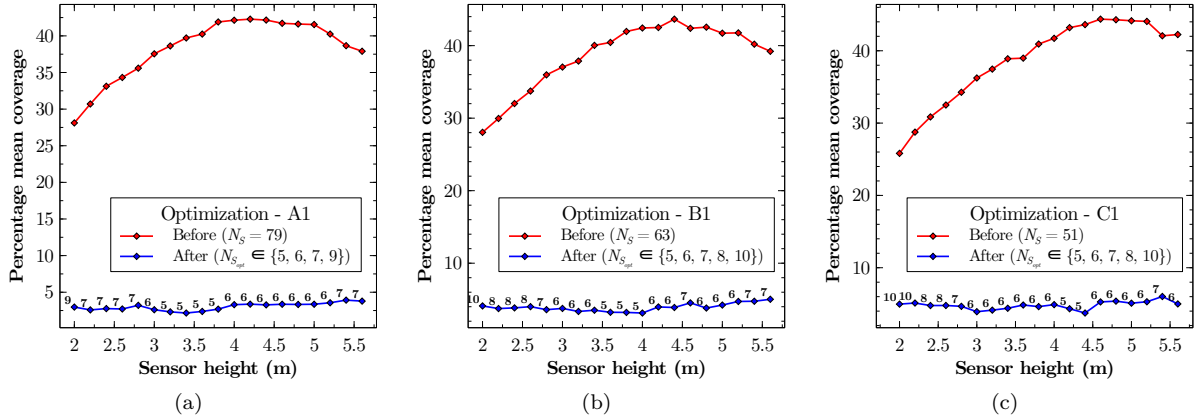


Fig. 7: Performance in terms of coverage of target points before and after optimization for sensor configurations:(a) A1, (b) B1, (c) C1, under different sensor heights

for optimizing placement of roadside infrastructure sensors using a regularized BIP-based approach. We have demonstrated the benefits of this framework through experimental results considering different sensor and environment configurations. We note that the framework is able to deliver very useful results when a sufficiently large number of candidate sensor locations are specified.

There are a number of assumptions considered to obtain an optimal solution, such as uniform type and installation height for all sensors and ensuring that the road network is topographically flat and geometrically linear. The coverage estimation is also based on a line-of-sight approach without sensor or environment physics. These are factors which may affect real-world sensor cov-

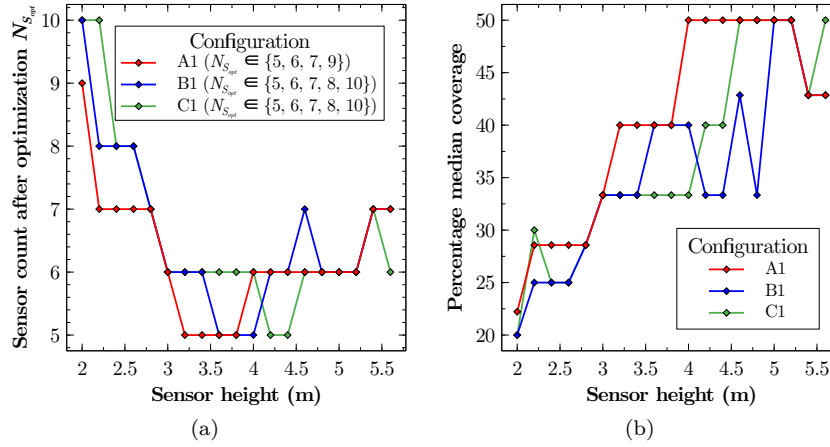


Fig. 8: Optimization performance: (a) Number of sensors selected after optimization,  $N_{S_{opt}}$  (b) percentage median coverage after optimization

erage. Therefore, adding the capability of handling curved and irregularly shaped road networks and working with high fidelity physics-based (multi-sensor/environment) simulations, can make the framework more robust.

We also assume that there is no upper limit on the count of optimal sensor locations. This may introduce the need to verify the sensitivity of the performance against the density and distribution of target points and sensors. Future versions of the framework can include an upper limit on maximum permissible sensors after optimization.

## REFERENCES

- [1] SAE, “J3016 standard: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” 2021.
- [2] Z. Liu, M. Arief, and D. Zhao, “Where should we place lidars on the autonomous vehicle?-an optimal design approach,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2793–2799.
- [3] J.-W. Ahn, T.-W. Chang, S.-H. Lee, and Y. W. Seo, “Two-phase algorithm for optimal camera placement,” *Scientific Programming*, vol. 2016, 2016.
- [4] J.-J. Gonzalez-Barbosa, T. García-Ramírez, J. Salas, J.-B. Hurtado-Ramos *et al.*, “Optimal camera placement for total coverage,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 844–848.
- [5] E. Horster and R. Lienhart, “Approximating optimal visual sensor placement,” in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1257–1260.
- [6] J. Urrutia, “Art gallery and illumination problems,” in *Handbook of computational geometry*. Elsevier, 2000, pp. 973–1027.
- [7] J. O’rourke, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.
- [8] J. Ferreira, A. Vale, and R. Ventura, “Optimizing range finder sensor network coverage in indoor environment,” *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 282–287, 2010.
- [9] R. Cole and M. Sharir, “Visibility problems for polyhedral terrains,” *Journal of Symbolic Computation*, vol. 7, no. 1, pp. 11–30, 1989.
- [10] M. Argany, F. Karimipour, F. Mafi, and A. Afghantloee, “Optimization of wireless sensor networks deployment based on probabilistic sensing models in a complex environment,” *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, 2018.
- [11] V. Akbarzadeh, J. C. Lévesque, C. Gagné, and M. Parizeau, “Efficient sensor placement optimization using gradient descent and probabilistic coverage,” *Sensors (Switzerland)*, vol. 14, no. 8, pp. 15 525–15 552, 2014.
- [12] F. Geissler and R. Gräfe, “Optimized sensor placement for dependable roadside infrastructures,” in *2019 Intelligent Transportation Systems Conference*. IEEE, 2019, pp. 2408–2413.
- [13] H. Tan, X. Hao, Y. Wang, F. C. M. Lau, and Y. Lv, “An Approximate Approach for Area Coverage in Wireless Sensor Networks,” *Procedia - Procedia Computer Science*, vol. 19, no. Ant, pp. 240–247, 2013.
- [14] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE transactions on computers*, vol. 51, no. 12, pp. 1448–1453, 2002.
- [15] J. Zhao, R. Yoshida, S.-c. S. Cheung, and D. Haws, “Approximate techniques in solving optimal camera placement problems,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, p. 241913, 2013.
- [16] H. Aghajan and A. Cavallaro, *Multi-camera networks: principles and applications*. Academic press, 2009.
- [17] X. Chen and J. Davis, “Camera placement considering occlusion for robust motion capture,” *Computer Graphics Lab, Stanford University, Tech. Rep.*, vol. 2, no. 2.2, p. 2, 2000.
- [18] N. Conci and L. Lizzi, “Camera placement using particle swarm optimization in visual surveillance applications,” in *2009 16th IEEE international conference on image processing (ICIP)*. IEEE, 2009, pp. 3485–3488.
- [19] Y. Morsly, N. Aouf, M. S. Djouadi, and M. Richardson, “Particle swarm optimization inspired probability algorithm for optimal camera network placement,” *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1402–1412, 2011.
- [20] S. Bianco and F. Tisato, “Sensor placement optimization in buildings,” in *Image Processing: Machine Vision Applications V*, vol. 8300. Int. Soc. for Optics & Photonics, 2012, p. 830003.
- [21] S. D. Roth, “Ray casting for modeling solids,” *Computer graphics & image processing*, vol. 18, no. 2, pp. 109–144, 1982.
- [22] Mathworks, “Mathworks roadrunner.” [Online]. Available: <https://mathworks.com/products/roadrunner.html>
- [23] KhronosGroup, “glTF Specifications.” [Online]. Available: <https://github.com/KhronosGroup/glTF>
- [24] Dawson-Haggerty *et al.*, “trimesh.” [Online]. Available: <https://trimsh.org/>
- [25] M. Berkelaar and J. Dirks, “lpsolve reference guide.” [Online]. Available: <http://lpsolve.sourceforge.net/5.5/>
- [26] C. B. Sullivan and A. A. Kaszynski, “Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk),” *Journal of Open Source Software*, vol. 4, no. 37, p. 1450, 2019.
- [27] M.-A. Mittet, H. Nouira, X. Roynard, F. Goulette, and J.-E. Deschaud, “Experimental assessment of the quanergy m8 lidar sensor,” in *ISPRS 2016 congress*, 2016.