

Where Should We Place LiDARs on the Autonomous Vehicle? - An Optimal Design Approach

Zuxin Liu¹, Mansur Arief² and Ding Zhao^{2,*}

Abstract—Autonomous vehicle manufacturers recognize that LiDAR provides accurate 3D views and precise distance measures under highly uncertain driving conditions. Its practical implementation, however, remains costly. This paper investigates the optimal LiDAR configuration problem to achieve utility maximization. We use the perception area and non-detectable subspace to construct the design procedure as solving a min-max optimization problem and propose a bio-inspired measure – volume to surface area ratio (VSR) – as an easy-to-evaluate cost function representing the notion of the size of the non-detectable subspaces of a given configuration. We then adopt a cuboid-based approach to show that the proposed VSR-based measure is a well-suited proxy for object detection rate. It is found that the Artificial Bee Colony evolutionary algorithm yields a tractable cost function computation. Our experiments highlight the effectiveness of our proposed VSR measure in terms of cost-effectiveness configuration as well as providing insightful analyses that can improve the design of AV systems.

I. INTRODUCTION

Critical to establishing safe driving for an autonomous vehicle (AV) is ensuring how rapidly and accurately it can perceive the surrounding environment and plan for maneuvers. Extensive efforts undertaken to improve AVs' perception include installing industry-grade cameras, such as Grashopper2, Flea2, etc. [1], [2], [3]. These efforts are highly dependent on the device configurations themselves, but even more on external environmental conditions, such as lighting. Therefore, many AV manufacturers have embraced LiDAR because it provides distance measures, thus reconstructing 3D views, and gives 360-degree coverage up to a reasonable distance. Some AV designs even incorporate multiple, expensive LiDARs to gain an even higher degree of seeing capability based on sensor redundancy [4]. Unfortunately, the market price of LiDARs remains substantial [5]. For example, a 16-beam Velodyne LiDAR costs almost \$8,000 [6].

To the best of our knowledge, the research on optimal, lower-cost LiDAR configurations for AVs is rare. A few studies have investigated how to achieve optimal configuration for the camera or 3D sensors, e.g., [7] proposed an optimal camera configuration method to achieve the largest field of view, while [8] optimize the camera's pose for motion capture systems. Both methods cannot be directly applied for optimal LiDAR configuration, because, compared to cameras, the receptive field of LiDAR is much more discrete.

Previously, in [9], we studied the LiDAR configuration problem by considering it as a min-max optimization and using a cylindrical representation as a proxy to the cost function and a mixed integer programming (MIP) to solve the model.

Motivated by the fact that our previous work suffers from the curse of dimensionality, i.e., it is unsuitable for solving large-scale problems, such as those involving multiple LiDARs, in this paper, we propose to reformulate the cost function by utilizing an easy-to-solve proxy. We use volume to surface area ratio (VSR), which has both geometric and bionic interpretations as a measure to evaluate the performance of a particular configuration. The resulting solution allows us to analyze the trade-offs between an AV's seeing capability and a LiDAR's design cost. Our goal is to find the right balance between the reliability and affordability of AV perception systems without sacrificing driving safety. Our proposed approach also allows us to tractably evaluate the complex sensors specifically designed for autonomous systems, such as artificial compound eyes [10], [11].

The remainder of this paper is as follows. Section II introduces the general concept and structure of our proposed approach to the optimal LiDAR configuration problem. Section III describes the model and our use of the Artificial Bee Colony (ABC) algorithm. Section IV explains the experiment settings and discusses the results. Section V concludes and suggests some future directions.

II. METHODOLOGICAL FRAMEWORK

In this section, we begin by defining the region of interest (ROI) and the perception area of LiDAR beams, both of which form the basis of the configuration problem. Then we introduce the non-detectable subspace and propose the maximum VSR as the cost function. We also adopt the Artificial Bee Colony (ABC) algorithm to solve the optimization problem.

A. ROI and perception area of LiDAR beams

We account for the limitation of LiDAR's detection range by considering that the LiDAR's placement does not influence the regions that are far away from the AV. This consideration aligns with many real-world applications where the perception algorithm chooses only a portion of the LiDAR sensor data as its input. For example, VoxelNet [12] use the point cloud data within a certain bounding box around the LiDAR as the input to perform the object detection task. Considering the ROI around the AV in the LiDAR configuration problem, we want to obtain as much

* Corresponding author: Ding Zhao. Email: (dingzhao@cmu.edu)

¹Zuxin Liu is with the School of Instrumentation Science and Optoelectronics Engineering, Beihang University, China

²Mansur Arief and Ding Zhao are with the Department of Mechanical Engineering, Carnegie Mellon University, USA.

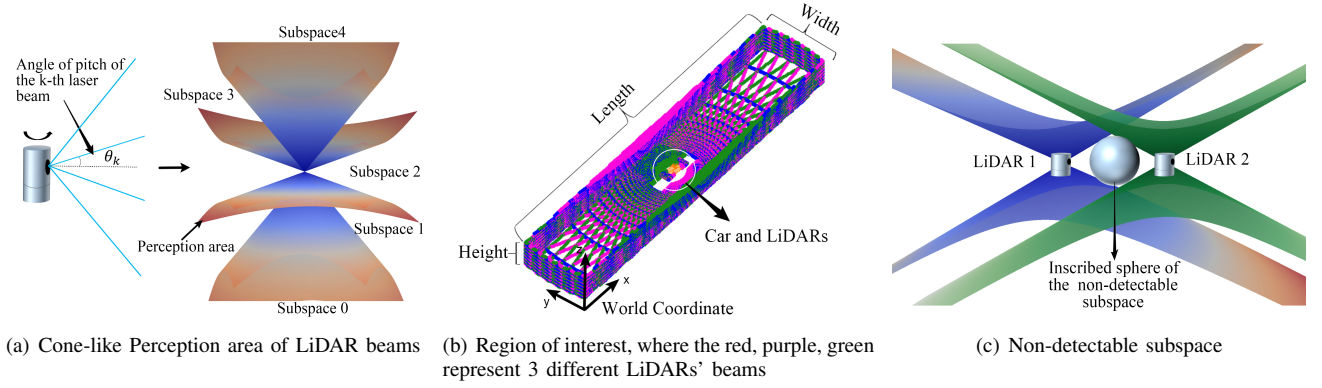


Fig. 1: Perception area, ROI and subspace demonstration

information as possible by well-covering the whole ROI with LiDAR beams.

We define the ROI by three-dimensional geometric properties – the length, width, and height – of a cuboid in the world coordinate. Fig. 1(b) shows an example of an AV located at the bottom-center of the x - y plane of the ROI, which allows us to define the perception area as the whole trajectory of LiDAR beams after a 360-degree rotation, i.e., each beam forms a surface. Note that we can only detect the objects that intersect with the perception area.

To simplify the representation, we assume that the beam always provides a perception line along its trajectory, i.e., the trajectory line forms a cone when the LiDAR beams rotate 360-degrees. Thus, we can model the whole perception area of a multi-beams LiDAR as the union of finitely many cones sharing the same vertex and vertical axis. Fig. 1(a) shows the cone-like perception area of the LiDAR beams, where θ_k represents the pitch angle of the k -th laser beam.

B. Non-detectable subspace segmentation

Given the ROI and the LiDAR configuration, the perception area of each LiDAR can intersect with each other as well as with the ROI boundaries. Therefore, the perception area segments the ROI into many irregular shape subspaces. We define them as non-detectable subspaces because the object located within one of the subspaces cannot be detected if both the LiDAR and object are static.

Fig. 1(c) shows an example of the non-detectable subspaces. We let N_l denote the number of LiDARs to be placed. Each LiDAR has N_b beams. Fig. 1(a) shows one LiDAR segmenting the ROI into $N_b + 1$ subspaces. From bottom to top, we denote each subspace by a number in ascending order from 0 to N_b . Therefore, for N_l LiDARs, the maximum number of subspace N_s is

$$N_s = (N_b + 1)^{N_l}. \quad (1)$$

We note that it is hard to represent the non-detectable subspaces analytically because their shapes are different and irregular. Section III introduces our solution, which is a cuboid-based segmentation method.

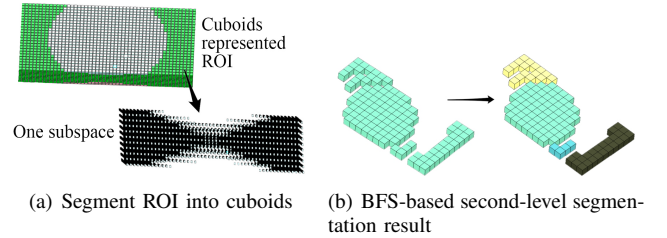


Fig. 2: Subspace segmentation

C. Cost function

Note that all of the non-detectable subspaces should be *as small as possible* to obtain the most informative perception from the LiDAR. Since the notions of how big or small are somewhat vague because each shape is different and irregular, we use the radius of the inscribed sphere to define the size of the subspace. As Fig. 1(c) shows that the subspace of the largest size, i.e., the one with the largest blindspot, is the one with the largest inscribed sphere.

To fix the worst possible case, we express the optimal LiDAR configuration as a min-max optimization problem to minimize the largest inscribed spheres from all subspaces. See [9] for the details. We note, however, that it is very hard to solve the inscribed sphere directly, so we propose another easy-to-solve indicator – the volume to surface area ratio (VSR) – which is inspired by bionics, to define the size of the subspace.

In the bionic world, VSR could be one dominating factor in the shapes and sizes of cells [13], [14]. A cell receives nutrition from the outside environment through the cell membrane; the surface area represents the cell's efficiency in absorbing nutrition, and its volume represents the amount of nutrition needed. As the cell grows bigger, its volume and membrane surface area also enlarge. Because the increase rate of the surface area is less than the volume, and the rate of absorption decreases, it is not economical for a cell to grow too big [13].

In this paper, we consider each non-detectable subspace as a cell and each surface area as the membrane. Loosely

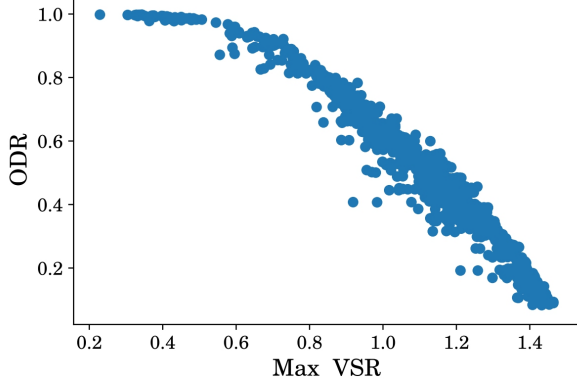


Fig. 3: General relationship between VSR and ODR

speaking, we consider the object to be detected as nutrition. Thus, we want to obtain the smallest possible VSR for optimal detection, especially of small objects. Specifically, we are interested in mitigating the worst-case scenario, i.e., we want to minimize the maximum VSR from the constructed non-detectable subspaces of the ROI.

We can also describe the VSR by using the formula of the radius of an inscribed sphere in a polyhedron [15]. When we know the volume, denoted as V , and the surface area of the polyhedron, denoted as S , we can express the radius of inscribed sphere R as $R = 3 \times \frac{V}{S}$. Therefore, the VSR can represent the ‘size’ of any shape of our interest.

In practice, engineers value the ability of LiDAR configurations to detect objects. In this paper, the object detected rate (ODR) represents the probability of an AV to detect a certain object. We let some objects of interest, which can also be represented as the cuboids-based objects, randomly appear within the ROI. We denote N_{ds} as the number of subspaces occupied by the cuboids. If $N_{ds} > \text{thres}$, where thres is a predefined threshold, we assume the object is detected. In the following experiment, we simulate the object to randomly appear within the ROI M times and tally the occurrences T where the object is detected. This setting allows us to calculate the ODR as $ODR = \frac{T}{M}$. Minimizing the maximum VSR will increase ODR (Fig. 3 is the general relationship between these measures). Therefore, we set the objective of the optimization model as minimization.

D. Artificial Bee Colony optimization algorithm

We note that the optimization problem is non-convex, and the solution space is continuous, high-dimensional and constrained. Therefore, using heuristic optimization method is a suitable choice in this case. We adopt the Artificial Bee Colony (ABC) optimization algorithm [16] as the solver for this LiDAR configuration problem, because its better performance and fewer hyper-parameters than other similar population-based algorithms [17].

The ABC optimization algorithm is inspired by the behavior of foraging bee swarms. The solutions correspond to food sources, and the cost function corresponds to the nectar

amount (i.e., the fitness) of a particular food source. Greedy selection policy is employed to ensure that the algorithm converges quickly, and a roulette wheel selection method is adopted to explore around the food sources for the nectar of the highest amount.

Technically speaking, the swarm of bees is divided into three groups: employed bees, onlookers, and scouts. The employed bees measure the nectar amount of food sources, and each employed bee corresponds to a food source. The onlookers adopt the roulette wheel selection method to select the existing food sources to be further explored. If the quality of an existing food source has not been improved within a certain number of iterations, its corresponding employed bee abandons the food source and becomes a scout to search for a new food source randomly. See [18] for a comprehensive review of ABC algorithms.

III. MODEL FORMULATION

As mentioned, it is very hard to represent the subspaces analytically, so we adopt a cuboid-based method for approximation. By segmenting (discretizing) the ROI into many small cuboids, we only need to determine which subspaces belong to which cuboids. Fig. 2 shows the steps: segment the ROI into many cuboids; transform each cuboid from world global coordinate to LiDAR local coordinate; perform first-level segmentation and use base- n notation to represent each subspace, and for each base- n notation use breadth-first search method to perform second-level segmentation.

A. Cuboid coordinate transformation

We denote θ_k as the k -th laser beam of a LiDAR, and formulate the cone-like perception area as:

$$z_p = \tan \theta_k \sqrt{x_p^2 + y_p^2}, \quad (2)$$

where x_p, y_p, z_p are the coordinate points on the perception area relative to the LiDAR coordinate system. Since it is difficult to formulate the cone in the world coordinate if the LiDAR rotates, we transform the cuboid from world coordinate into LiDAR coordinate to facilitate the following segmentation.

Let $[x_w, y_w, z, \alpha_w, \beta_w, \gamma_w]$ denote the pose of a LiDAR in the world coordinate, where $[x_w, y_w, z]$ represent the position and $[\alpha_w, \beta_w, \gamma_w]$ represent the orientation. Then we obtain the rotation matrix $R_{\alpha\beta\gamma}$ as

$$R_{\alpha\beta\gamma} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}, \quad (3)$$

where c represents \cos and s represents \sin in short. For example, c_α represents $\cos(\alpha)$. We denote $T_{xyz} = [x, y, z]^T$ as the translation vector, and express the transformation matrix from the LiDAR coordinate to the world coordinate T_{wl} as

$$T_{wl} = \begin{bmatrix} R_{\alpha\beta\gamma} & T_{xyz} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (4)$$

Finally, we denote $\mathbf{X}_w = [x_w, y_w, z_w]$ and $\mathbf{X}_l = [x_l, y_l, z_l]$ as the coordinates of the cuboids in the world coordinate

system and LiDAR coordinate system, respectively. Then, the transformation between the coordinate system satisfies

$$\mathbf{X}_l = \mathbf{T}_{wl}^{-1} \mathbf{X}_w. \quad (5)$$

The two systems, which represent the cuboids in both the LiDAR and the world systems, set the stage for the two segmentation processes, which we decompose into two levels: base-n first-level segmentation and breadth-first search (BFS) second-level segmentation.

B. Base-n notation first-level segmentation

After obtaining the cuboid coordinates in the LiDAR coordinate, the next step is to determine which subspace the cuboid belongs to. Here, we use base-n notation to identify the subspace.

We assume there are N_l N_b -beam LiDARs to be placed. According to equation (1), each LiDAR could segment the ROI into $N_b + 1$ subspaces, resulting in $(N_b + 1)^{N_l}$ maximum number of subspaces. Therefore, each subspace can be represented by a base-n notation $d_1 d_2 \dots d_i \dots d_n$, where n equals the LiDARs' number N_l , and d_i indicates the number of subspaces of the i -th LiDAR, ranging from 0 to N_b .

We denote θ_{ik} as the pitch angle of the k -th beam of the i -th LiDAR, where $k \in 0, 1, \dots, N_b$. $[x_{il}, y_{il}, z_{il}]$ is the cuboid coordinates in the i -th LiDAR coordinate system. To determine the digit d_i , we need to find the unique beam k that satisfies

$$\begin{cases} z_{il} \geq \tan \theta_{i(k-1)} \sqrt{x_{il}^2 + y_{il}^2}, \\ z_{il} < \tan \theta_{ik} \sqrt{x_{il}^2 + y_{il}^2} \end{cases} \quad 1 \leq k \leq N_b - 1 \quad (6)$$

or

$$z_{il} < \tan \theta_{ik} \sqrt{x_{il}^2 + y_{il}^2}, \quad k = 0 \quad (7)$$

or

$$z_{il} \geq \tan \theta_{ik} \sqrt{x_{il}^2 + y_{il}^2}, \quad k = N_b \quad (8)$$

Then, the digit $d_i = k$ and thus, iterating over all the LiDARs allow us to determine the cuboids' subspaces in the ROI.

C. Breadth-first search (BFS) based second-level segmentation

A base-n notation of the subspace denotes each cuboid in the ROI. The cuboids of the same base-n notation are the non-detectable subspace segmented by the laser beams. However, the segmentation is not thorough. For example, the right side of Fig. 2(b) shows the cuboids of the same base-n notation. Therefore, we use a BFS-based method to perform the second-level segmentation.

We classify the cuboid that has an overlapping face with the root node into the same group of the root cuboid. After searching all the non-visited candidate cuboids, we mark the root node cuboid as visited. We implement the same process for the left non-visited candidate cuboids until all the cuboids are marked as visited. The left side of Fig. 2(b) shows the result of performing the second-level segmentation.

D. Optimization model

To construct the optimization problem, we denote the number of LiDARs as N_l and the number of laser beams as N_b . We denote the i -th LiDAR configuration as $C_i = [x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i]$, where $\alpha_i, \beta_i, \gamma_i$ each represents the yaw, pitch and roll of the LiDAR in world coordinate, respectively.

Since the LiDAR configuration C is limited by the installation position of the car, we define $C_{\min} = [x_{\min}, y_{\min}, z_{\min}, \alpha_{\min}, \beta_{\min}, \gamma_{\min}]$ and $C_{\max} = [x_{\max}, y_{\max}, z_{\max}, \alpha_{\max}, \beta_{\max}, \gamma_{\max}]$ which represent some lower- and upper-bound of the LiDAR configurations, respectively. As such, the i -th configuration C_i is valid if it satisfies the physical constraint:

$$C_{\min} \leq C_i \leq C_{\max}. \quad (9)$$

Let $CN = \{C_1, C_2, \dots, C_{N_l}\}$ be the configuration set. Further, if we denote the VSR of the j -th subspace as $Loss_j(CN)$ for any given configuration set CN and denote the set of cuboids coordinate in j -th subspace as S_j , then

$$Loss_j(CN) = \frac{V(S_j)}{SA(S_j)}, \quad (10)$$

where $V(S_j)$ and $SA(S_j)$ are the volume and surface area of the S_j cuboid set, respectively. Solving the $V(S_j)$, only requires calculating the number of the cuboids. We denote $[e_x, e_y, e_z]$ as the cuboids edges length along the x, y, z axis such that the number of cuboids in S_j is n_j . We solve $V(S_j)$ by:

$$V(S_j) = e_x e_y e_z n_j \quad (11)$$

The surface area $SA(S_j)$ comprises 3 parts: the surface area along $x - y$ plane SA_{xy} , along $x - z$ plane SA_{xz} and along $y - z$ plane SA_{yz} . To know the size of the cuboid, we only need to count the overlap faces along the different planes.

The algorithms to solve the 3 parts are the same, so here we use SA_{xy} as an example. We denote S as the cuboid set, and S_{ix}, S_{iy}, S_{iz} as the i -th cuboid's index along the x, y, z axes. The steps are shown in Algorithm 1.

We formulate the optimization problem as

$$\begin{aligned} \overline{CN} = \arg \min_{CN} \max_j & \quad Loss_j(CN) \\ \text{subject to} & \quad C_{\min} \leq CN \leq C_{\max}, \end{aligned} \quad (12)$$

where \overline{CN} represents the optimized configuration solution.

To solve (12), we deploy the ABC algorithm and generate a swarm of artificial bees to explore the solution space. Let $x_i = (x_{i1}, \dots, x_{i2}, \dots, x_{id})$ ($i = 1, 2, \dots, \tau$) denote the solutions, where τ represents the number of bees and d represents the solutions dimension. $fit(x_i)$ represents the quality of the solution x_i , which is negatively related with our cost function. Defining probability P_i as

$$P_i = \frac{fit(x_i)}{\sum_{i=1}^{\tau} fit(x_i)}, \quad (13)$$

Algorithm 1 Solving the Surface Area of the Cuboid Set

Require: Cuboid set, S ; Cuboid resolution, $[e_x, e_y, e_z]$;
Ensure: The surface area along $x - y$ plane, SA_{xy} ;

- 1: Sort the coordinates of cuboids in S according to $x - y - z$ axes in an increasing order
- 2: $sz = \text{Size}(S)$, $cnt = 0$, $i = 1$
- 3: **repeat**
- 4: **if** $S_{ix} = S_{(i-1)x}$ and $S_{iy} = S_{(i-1)y}$ and $S_{iz} = S_{(i-1)z} + 1$ **then**
- 5: $cnt = cnt + 1$
- 6: **end if**
- 7: $i = i + 1$
- 8: **until** $i = sz$
- 9: $SA_{xy} = 2(sz - cnt)e_x e_y$

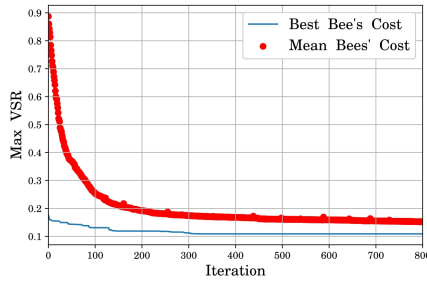


Fig. 4: The convergence trend of ABC solutions for minimizing the cost function - the max VSR

allows the onlookers to select the solution x_i and examine its quality $fit(x_i)$. Next, the algorithm prescribes a new solution v_i near the currently selected solution x_i by computing

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad (14)$$

where $k \in \{1, 2, \dots, \tau\}$ and $j \in \{1, 2, \dots, d\}$ are randomly chosen indices. We note that ϕ_{ij} is a random number in the range $[-1, 1]$. If the $fit(v_i) > fit(x_i)$, then we replace the solution x_i with v_i , i.e., the bees move to v_i . Adopting the analogy described in Section II-D, Algorithm 2 summarizes the iterative steps of the ABC algorithm.

IV. EXPERIMENT AND DISCUSSION

We consider an ROI of size $[60, 20, 4]$ in meters, excluding the rectangular region $x \in [27, 33], y \in [8, 12], z \in [0, 4]$, which is very close to the AV and we do not consider its in the perception algorithm at this point. The lower- and upper- bound of the LiDARs pose are $[28, 9, 2.2, 0, 0, 0]$ and $[31, 11, 3, 3.1415, 3.1415, 0]$, where the first three components are measured in meters and the last three are measured in radians. The *yaw* angle is not be optimized because the LiDAR will rotate 360 degrees.

We use Velodyne VLP-16 [6] as our candidate LiDAR components. For tractability, we set the resolution of the cuboids as $[1, 0.5, 0.2]$. To solve the problem, we set 200 bees and use the ABC algorithm to iterate 800 times. Fig. 4 shows how the ABC algorithm converges in the 4 LiDARs'

Algorithm 2 Artificial Bee Colony (ABC) Algorithm

Require: Number of bees, τ ; Iteration number, $IterMax$;
Threshold to abandon a food source, $thres$;

Ensure: The optimal solution, x_i , $i = 1, \dots, \tau$;

- 1: Randomly initialize the food source x_i , $i = 1, \dots, \tau$
- 2: Get the quality of each food source $fit(x_i)$
- 3: Set $cycle = 0$ and $cnt_i = 0$, $i = 1, \dots, \tau$
- 4: **repeat**
- 5: **for** each food source x_i **do**
- 6: Generate new food source v_i using equation (14)
- 7: **if** $fit(v_i) > fit(x_i)$ **then**
- 8: Replace x_i with v_i
- 9: **else**
- 10: $cnt_i = cnt_i + 1$
- 11: **end if**
- 12: **end for**
- 13: **for** each onlooker **do**
- 14: Select a food source using a roulette wheel selection method based on equation (13)
- 15: Generate new food source v_i around x_i using equation (14)
- 16: **if** $fit(v_i) > fit(x_i)$ **then**
- 17: Replace x_i with v_i
- 18: **else**
- 19: $cnt_i = cnt_i + 1$
- 20: **end if**
- 21: **end for**
- 22: **for** each food source x_i **do**
- 23: **if** $cnt_i = thres$ **then**
- 24: Replace x_i with a randomly generated solution
- 25: **end if**
- 26: **end for**
- 27: $cycle = cycle + 1$
- 28: **until** $cycle$ equals $IterMax$

case. The red line represents the average value of all the bees' max VSR, and the blue line represents the best solution's max VSR in each iteration. Fig. 4 shows that the search area of the bees converges to the regions associated with lower cost values. Fig. 5 shows the optimized LiDAR configuration solutions for 1, 2, 3, and 4 LiDARs.

A. Maximum VSR trend in the number of LiDARs

Given the type of LiDAR, AV designers may need to determine a proper number of LiDARs. Fewer LiDARs may influence the detection performance, but too many LiDARs causes information redundancy and increases the computation burden. We showcase an application for determining the number and type of LiDARs to place. We consider three types of LiDARs: 4-beam, 8-beam, 16-beam LiDAR, with evenly distributed angles of the pitch from -15° to 15° . Our goal is to yield the optimal configuration and compare the detection uncertainty performance for the different types of LiDARs.

Fig. 6 shows the results after 800 iterations. The fact that the marginal improvement rate decays as the number

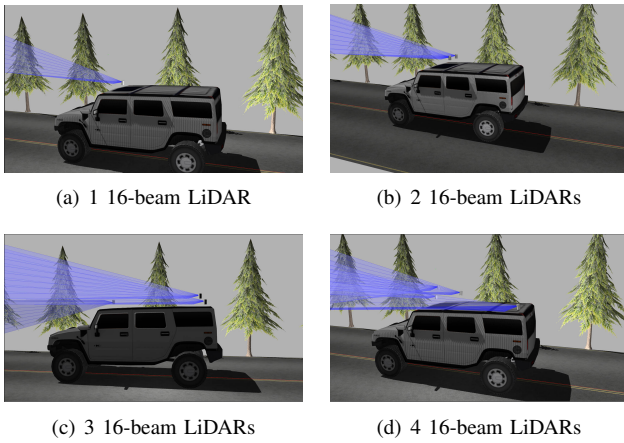


Fig. 5: LiDAR configuration visualization

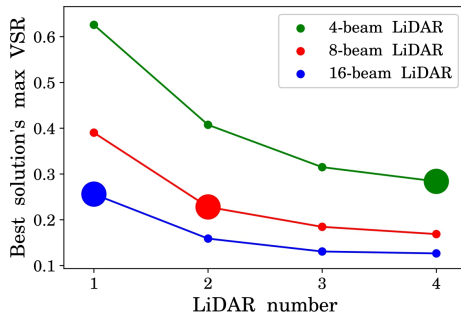


Fig. 6: Best solution's maximum VSR of different LiDAR configuration after 800 iterations

of LiDARs increase implies that the LiDAR beams space configuration, which can also be interpreted as the ability to detect small objects, may not increase when the number of LiDARs reaches a certain value. Optimal configuration plays a significant role when an AV design employs a few sensors, such as the artificial compound eye [11]. The fact that the maximum VSR of the AV design is either larger, or the same at best, when the configuration is not optimized, implies a dominated performance compared to the solution obtained by the proposed method.

B. Comparison between different LiDAR types

The proposed method can help AV engineers choose the optimal LiDAR types, or improve LiDAR design. For example, the marked points in Fig. 6 show that two 8-beam LiDARs outperform both the one 16-beam and the four 4-beam LiDARs *when configured optimally using our proposed method*. Intuitively, the four 4-beam LiDARs should be chosen, because its configuration is more flexible. However, its physical constraints, i.e., the angles of the pitch of its beams, can limit performance. This insight analysis will be highly advantageous for AV designer to better achieve affordable and safe AV design, and for LiDAR manufacturer better design the LiDAR type.

V. CONCLUSIONS

This paper proposed a solution to the optimal LiDAR configuration problem which will help designers and manufacturers determine the proper number of sensors and their placement on autonomous vehicles. The overall relationship between the size of non-detectable subspaces, such as the blindspot of a given LiDAR configuration, the volume to surface area ratio (SVR), and the object detection rate (ODR) was investigated by using a new VSR-based measure. The Artificial Bee Colony (ABC) algorithm was adopted to segment the subspaces of the region of interest (ROI) into a cuboid-based representation. That yielded efficient computational performance. The experiment results indicated the effectiveness of the solution in prescribing configurations that obtained maximum detection performance without becoming computationally burdensome.

The research in this paper will be extended to investigate occlusion problems, and the solutions will be validated with real-world testing. The results are expected to bring AV designers, engineers, and manufacturers together to deploying autonomous vehicles and fleets with a reliable and affordable perception system.

REFERENCES

- [1] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, vol. 2, 2014, p. 3.
- [2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [3] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image and Vision Computing*, vol. 68, pp. 14–27, 2017.
- [4] B. Schwarz, "Lidar: Mapping the world in 3d," *Nature Photonics*, vol. 4, no. 7, p. 429, 2010.
- [5] N. Jayaweera, N. Rajatheva, and M. Latva-aho, "Autonomous driving without a burden: View from outside with elevated lidar," *arXiv preprint arXiv:1808.08617*, 2018.
- [6] Velodyne. [Online]. Available: <https://velodynelidar.com/vlp-16.html>
- [7] J. Dybedal and G. Hovland, "Optimal placement of 3d sensors considering range and field of view," in *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1588–1593.
- [8] J. E. Banta and M. A. Abidi, "Autonomous placement of a range sensor for acquisition of optimal 3-d models," in *Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on*, vol. 3. IEEE, 1996, pp. 1583–1588.
- [9] S. Mou, Y. Chang, W. Wang, and D. Zhao, "An optimal lidar configuration approach for self-driving cars," *arXiv preprint arXiv:1805.07843*, 2018.
- [10] D. Floreano, R. Pericet-Camara, S. Viollet, F. Ruffier, A. Brückner, R. Leitel, W. Buss, M. Menouni, F. Expert, R. Juston, et al., "Miniature curved artificial compound eyes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 23, pp. 9267–9272, 2013.
- [11] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, p. 460, 2015.
- [12] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *arXiv preprint arXiv:1711.06396*, 2017.
- [13] W. M. Lewis, "Surface/volume ratio: implications for phytoplankton morphology," *Science*, vol. 192, no. 4242, pp. 885–887, 1976.
- [14] L. K. Harris and J. A. Theriot, "Surface area to volume ratio: A natural variable for bacterial morphogenesis," *Trends in microbiology*, 2018.

- [15] E. Weisstein. Insphere, from mathworld—a wolfram web resource. [Online]. Available: <http://mathworld.wolfram.com/Insphere.html>
- [16] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm,” *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [17] D. Karaboga and B. Akay, “A comparative study of artificial bee colony algorithm,” *Applied mathematics and computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [18] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: artificial bee colony (abc) algorithm and applications,” *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.