# arXiv:2403.05477v1 [cs.RO] 8 Mar 2024

# Take Your Best Shot: Sampling-Based Next-Best-View Planning for Autonomous Photography & Inspection

Shijie Gao\*1, Lauren Bramblett\*2 and Nicola Bezzo<sup>1,2</sup>

Abstract—Autonomous mobile robots (AMRs) equipped with high-quality cameras have revolutionized the field of inspections by providing efficient and cost-effective means of conducting surveys. The use of autonomous inspection is becoming more widespread in a variety of contexts, yet it is still challenging to acquire the best inspection information autonomously. In situations where objects may block a robot's view, it is necessary to use reasoning to determine the optimal points for collecting data. Although researchers have explored cloud-based applications to store inspection data, these applications may not operate optimally under network constraints, and parsing these datasets can be manually intensive. Instead, there is an emerging requirement for AMRs to autonomously capture the most informative views efficiently. To address this challenge. we present an autonomous Next-Best-View (NBV) framework that maximizes the inspection information while reducing the number of pictures needed during operations. The framework consists of a formalized evaluation metric using ray-tracing and Gaussian process interpolation to estimate information reward based on the current understanding of the partially-known environment. A derivative-free optimization (DFO) method is used to sample candidate views in the environment and identify the NBV point. The proposed approach's effectiveness is shown by comparing it with existing methods and further validated through simulations and experiments with various vehicles.

*Note*—Code and videos of the simulations and experiments are provided in the supplementary material and can be accessed at https://www.bezzorobotics.com/sg-lb-iros24.

# I. INTRODUCTION

Autonomous mobile robots (AMRs) are becoming increasingly common in a wide range of industries, including manufacturing, healthcare, logistics, and entertainment. Because of their agility and mobility, autonomous robots have been particularly useful for photography [1] and inspection purposes [2]. The deployment of robots for these applications also offers other benefits, including increased safety, efficiency, and accuracy. Robots equipped with cameras, depth sensors, and other tools can access areas that are hazardous or inaccessible to human inspectors [3] (e.g., disaster sites, high-rises, internal pipe systems, and underwater structures). The use of robots to inspect buildings and other infrastructures is a promising solution to address the challenges associated with traditional inspection methods. However, most of the existing autonomous inspection frameworks are not fully automated and collect all available information during operations, relegating data processing and decisionmaking to humans. Thus, to fully automate robotic inspection, robots must be able to assess the quality of inspection photos and discard unnecessary data. However, the evalua-



Fig. 1. A pictorial depiction of the problem. A UAV is performing an inspection task to capture an oblique view (i.e., front view) of a house. When comparing these four examples, a bird's eye view in D preserves the front view and provides the most comprehensive information.

tion of photos is inherently subjective. Even among human observers, reaching a unanimous agreement on the quality rating of a photo can be challenging [4], [5]. This introduces additional challenges in constructing a formal metric that precisely defines the criteria for a "good" picture. With these considerations in mind, we identify two notable gaps in the robotics literature focused on inspection and photography: i) the absence of an explicit definition of a good picture, and ii) that current inspection frameworks do not incentivize efficient data collection during prolonged tasks, overlooking the cost of consuming repetitive information and the need for extensive human post-processing efforts. We propose that bridging these gaps will make autonomous inspection more useful and accessible to real-world operations.

Typically, during inspection tasks, the location and dimensions of the region of interest (ROI) are known, and the surroundings of the ROI are uncertain. For example, in the house inspection shown in Fig. 1, the location of the building is known, while the landscape and adjacent structures can change, potentially obstructing the ROI and jeopardizing the safety of the unmanned aerial vehicle (UAV). In this scenario, the UAV should autonomously detect that trees obstruct viewing point (A) and that, by viewing from either side (B, C), more information can be obtained. The UAV's objective is to autonomously evaluate different viewpoint candidates without visiting them, identify the best viewpoint (a bird'seye view, D in the example in Fig. 1), and route to a new viewpoint, considering the observed obstacles.

We consider the framework presented in this paper as a Next-Best-View (NBV) solution in which the goal is to obtain the fewest viewing points and maximize inspection information. We propose an evaluation metric based on the perspective distortion, the scale of a target within the viewing frame, and the estimated target coverage which will allow us to find the best viewpoint. The metric is used in conjunction with a derivative-free optimization (DFO) method which

<sup>\*</sup> Co-first authors. Shijie Gao is the corresponding author.

Shijie Gao, Lauren Bramblett, and Nicola Bezzo are with the Departments of Electrical & Computer Engineering<sup>1</sup> and Systems & Information Engineering<sup>2</sup>, University of Virginia, Charlottesville, VA 22904, USA. Email: {sg9dn, gbr5kx, nb6be}@virginia.edu

samples the environment to find the best inspection point. The main contribution of this work is a computationally efficient Gaussian process interpolation using DFO to optimize the proposed evaluation metric over uniformly sampled candidate views maximizing target coverage at runtime.

# II. RELATED WORK

A well-studied problem in the field of mobile robot inspection is the Next-Best-View problem. First introduced by Connolly [6], NBV focuses on developing algorithms to efficiently plan and execute the exploration of objects or environments. The main goal of NBV is to select the most informative viewpoint for the robot to observe and improve its knowledge of a mission objective, such as an environment or an object. This is achieved by computing an optimized viewpoint based on a task-oriented optimization term that considers the robot's objectives and constraints. As NBV focuses on determining the optimal viewpoint for the next observation, it can be considered as a subset or specific case within the exploration framework. In the literature, NBV has been extensively explored for 3D object reconstruction. One of the challenges in these applications is how to choose the NBV. Authors in [7] employ a weighted multi-objective optimization approach to select NBVs for a mobile robotic arm, while [8] uses a double branch network architecture to rank NBVs. Although many applications involve a single robot, [9] leverages two robotic arms and multi-agent systems, respectively, to select NBVs that promote efficiency and achieve faster reconstruction. Although these works demonstrate the effectiveness of NBV in object reconstruction, they mainly focus on dense data collection, encouraging robots to gather data close to the target without considering repetitive data collection or the overall aesthetics of the target.

Another challenge of applying the NBV problem is computing the optimal viewpoint in partially known or unknown environments, where unexpected objects can not only obscure the target but also pose safety threats to vehicles. Although some research focuses on navigating robots to optimal viewpoints in known environments [10], several works address this challenge in conjunction with NBV. Authors in [11] investigate a receding-horizon NBV, utilizing a random tree method to guide the robot along a path in an unknown environment. [12] proposes a guided NBV approach for large-scale 3D reconstruction, which requires a rough global scan prior to a detailed NBV inspection. [13] computes NBV based on the map built during frontier exploration. Other works related to the NBV focus on exploring unknown environments [14]. In our work, the robot starts with the target's location as a priori information and dynamically updates the NBV to efficiently adapt to changes in the environment, ensuring effective guidance of the vehicle.

### **III. PROBLEM FORMULATION**

For inspection tasks, it is usually the case that the target information is known (i.e., the location of a house, pipeline, or structure) while the surroundings can be uncertain. In this work, we assume that the robot knows the target information,  $\mathbf{T}$ , including the location and dimensions. Obstacles in the environment, denoted as  $\mathcal{O}$ , can either be known or unknown and are represented by an occupancy map  $\mathcal{M}$ . To find the

best point for inspection, the vehicle should update the best view position as obstacles are detected that compromise inspection photo quality. We assume that the vehicle is equipped with a range sensor (i.e., LiDAR, sonar, cameras) which enables the robot to recognize obstacles and measure its distance to the target within the vehicle's field of view. The goal is to maximize the inspection information while reducing the number of pictures needed during operations. To achieve this, we formally define the problems as follows:

**Problem 1.** Metrics of the Best View for Target Inspection: Consider the location and dimensions of the target represented by the coordinate set  $\mathbf{T}$  and a set of initially unknown obstacles that are discovered at runtime and represented by an occupancy map  $\mathcal{M}$ . The goal is to find an evaluation metric  $G(\mathbf{T}, \mathcal{M}, \mathbf{P})$  that scores the sampled viewpoints  $\mathbf{P}$ for the vehicle to visit at runtime.

**Problem 2.** *Max-Info Path Planning:* Given a partiallyknown environment and metrics for picture evaluation, generate a path that minimizes the time to the inspecting task while ensuring the robot's safety. The path planner should also accommodate changes in the environment, as new obstacles may appear as the vehicle approaches the best inspection point.

# IV. METHODOLOGY

The proposed framework aims to provide a comprehensive strategy to inspect a target with the fewest number of viewpoints. In defining such an inspection viewpoint, the target object should occupy a certain proportion of the frame from the viewpoint's perspective, minimizing distortion and reducing any obstructions. If the target is not fully visible in one frame, the robot must decide autonomously to focus on one part of the target and subsequently move to capture the remainder of the target. In this work, we propose an objective function that mathematically defines what a quality inspection photo is and a method to capture quality images of the entire target with as few pictures as possible. Referring again to Fig. 1, we point out that the house is best viewed from viewpoint D since it is minimally obscured and distorted, while other views show little of the house or have a distorted view of the target.

Fig. 2 shows an overview of our approach to guide the autonomous mobile robot during the mission. As shown, the vehicle is initialized at a starting point in a partially-known environment with the target information known. The vehicle updates its map using its equipped depth sensor. After the update, candidate views are generated and evaluated via our quality scoring metrics. We use particle swarm optimization (PSO) to iteratively search for the best viewpoint in the environment but any DFO method can be utilized. The best viewpoint is then passed to the vehicle's planner and controller to drive it toward the location. As the vehicle moves, the best viewpoint is reassessed with any new information, given repeated map updates. Once the vehicle reaches the best viewpoint, a photo is captured and the target interest is updated based on the estimated target coverage from the captured photo using Gaussian process interpolation. Our approach is discussed in detail in the following sections, starting with the photo quality metrics followed by candidate view generation and evaluation.



Fig. 2. Diagram of the proposed approach.

# A. Photo Evaluation Metric

First, we introduce the metric to determine the ideal position to capture an image of a desired target. To aid the reader's comprehension, we begin with a visual illustration of the metric using a two-dimensional example in Fig. 3. Furthermore, we offer a similar three-dimensional example in Fig. 4 that showcases the expansion of the two-dimensional metrics to three dimensions. We consider a discretized target  $\mathbf{T} \in \mathbb{R}^{m \times n_d}$  where  $n_d$  is the dimension of the target and defined by m coordinates. In the 2D example, we show a UAV positioned slightly behind an obstacle, obscuring its view of the full target. For the visible portion of the target  $\mathbf{T}_v \subseteq \mathbf{T}$  and for  $\tau \in \mathbf{T}_v$ ,  $g(\tau, \boldsymbol{p}) = 1$  indicates that the visible portion has not been inspected previously and is visible from viewpoint p, while  $q(\tau, p) = 0$  indicates that the region is obscured or has already been observed in the past. In our approach, we seek to autonomously navigate and find the optimal location or a few locations to fully inspect the desired object. We define the optimal viewpoint  $p^*$  as follows:

$$\boldsymbol{p}^* = \underset{\boldsymbol{p}}{\operatorname{arg\,max}} \quad G(\mathbf{T}, \mathcal{M}, \boldsymbol{p})$$
 (1)

where 
$$G(\mathbf{T}, \mathcal{M}, \boldsymbol{p}) = \gamma_d \cdot \gamma_s \cdot \sum_{\tau \in \mathbf{T}} g(\tau, \boldsymbol{p}).$$
 (2)

The function  $g(\tau, p)$  signifies the level of interest that remains for any portion of the target after viewing from viewpoint p. As shown in Fig. 3, the visible portion of the target is a function of the position p and the occupancy map  $\mathcal{M}$ . The variables  $\gamma_d$  and  $\gamma_s$  are discount factors associated with utility functions  $U_d$  and  $U_s$  to assess the quality of a viewpoint given its respective camera parameters, position, and environmental obstructions. These factors are essential in determining the robot's choice of optimal viewpoints, as they influence the amount of distortion that can be accepted in the inspection pictures and the amount of the target that should occupy the image. The evaluation of these factors will be discussed in more detail in the following sections.



Fig. 3. Pictorial depiction of the evaluation metric for a 2D space.



Fig. 4. Pictorial depiction of evaluation metric for a 3D space.1) Perspective Distortion Discount Factor

Directly facing an inspected surface is often the ideal view for inspection tasks to maximize the amount of information in one photograph and create a consistent view of the target [15]. In fields such as product photography, architecture, or real estate where precise measurements and representations are required, if the view is at an angle, structures can appear distorted, which may compromise the resulting image capture. We introduce a *perspective distortion* discount factor that penalizes views that cause a distortion of the target. The target distortion in our approach is formally defined as follows:

$$\gamma_d = \prod_{i=1}^{n_d - 1} U_d \left( \frac{|\ell_{i2} - \ell_{i1}|}{\ell_{i\mathbf{T}}}, \mathbf{q}_d \right)$$
(3)

where  $\ell_{i1}$  and  $\ell_{i2}$  represent the maximum distances from the optical center associated with the intersection of the camera frame and the target. The variable  $\ell_{iT}$  is the length of the target captured along dimension i.  $U_d$  represents the user-defined utility function associated with the perspective distortion, penalizing unbalanced target captures, and  $q_d$  is a vector of parameters for the user-defined function. For example, in a 2D scenario if  $\ell_{11} = \ell_{12}$ , the view of the vehicle's camera is perpendicular to the target, minimizing perspective distortion and maximizing  $\gamma_d$ . If  $\ell_{11} >> \ell_{12}$ or  $\ell_{11} \ll \ell_{12}$ , the vehicle is positioned far to the left or right of the target, respectively, maximizing the perspective distortion and resulting in a small  $\gamma_d$ . This factor is designed to account for information loss that may occur in side views due to the inherent limitations of perspective projection. By incorporating this factor into the optimization process, the proposed method can effectively balance the trade-off between the ideal view and the available view, improving the overall quality of inspection images.

# 2) Scale Discount Factor

The term *scale* in photography refers to the relative size of the target versus all other objects in the frame. We define a favorable picture as a picture in which the target is in the foreground and the target occupies a percentage  $\beta$  of the frame. Accurate sizing of a subject within the frame is essential for an accurate representation of the target. If the scaling is incorrect, the image quality will be poor and there will be little useful information. An example of this is in real estate applications, where a properly sized image will give prospective buyers an idea of the layout of the property and the design elements of the home. We use  $\gamma_s$ to represent the scale discount factor that encourages an inspection image with a percentage of the frame being the target of interest. The penalty factor is determined using the target's estimated area within the frame. The variable  $\beta$  is a user-defined parameter that specifies the desired optimal percentage of the target that is within the frame of the picture. By applying a discount to target coverage using the scale discount factor, we ensure that the optimal image captured by the robot contains an appropriate and desirable proportion of the target object. We use the following equation to calculate this factor:

$$\gamma_s = \prod_{i=1}^{n_d-1} U_s\left(\frac{(\phi_{i\max} - \phi_{i\min})}{\Phi_i}, \beta, \mathbf{q}_s\right)$$
(4)

where  $\phi_{i \min}$  and  $\phi_{i \max}$  are the minimum and maximum angles of the intersection between the target plane and center of the frame along dimension *i* and  $\Phi_i$  is the field of view of the camera along dimension *i*. We denote  $U_s$  as the userdefined utility function associated with the scale of the target in the frame and  $\mathbf{q}_s$  as a vector of parameters for the userdefined function. In (4) , if  $\phi_{i \min} = -\Phi/2$  and  $\phi_{i \max} = \Phi/2$ , the target would occupy 100% of the frame along dimension *i*. As a result, the utility function  $U_s(\cdot)$  would score this percentage based on the parameter  $\beta$ . In this case, if  $\beta = 1$ , the value of  $\gamma_s$  would be maximized. To illustrate this concept, Fig. 3 presents a physical representation of the  $\phi$ s and the corresponding scale of the target within the vehicle's frame.

### B. Candidate View Evaluation

The criteria for an optimal inspection picture from the previous sections can be used to assess the quality of any viewpoint given the vehicle's current knowledge of the environment. As assumed in Sec. III, the vehicle is equipped with a depth sensor (e.g. lidar, stereo camera, RGB-D) and is capable of updating the occupancy map of the environment  $\mathcal{M}$ . In this application, the occupancy map is updated using recursive Bayesian updates [16], although any method can be used. The occupancy map ensures that as the robot moves through the environment, obstacles and obstructions are stored and planned for in subsequent iterations using  $\mathcal{M}$ . In turn, this requires that a view be reevaluated, since the quality of a view may increase or decrease as new occupied areas are observed. Thus, the quality of a viewpoint must be assessed at runtime; however, determining the vehicle's view of the target in a cluttered, partially-known environment is complex and potentially creates a discontinuous or highly non-linear solution space for (1). To reduce the complexity of the computation, we use ray-casting to determine the quality of a viewpoint.

Ray-casting is a classical technique in computer vision that allows complex interactions and quality estimations to be discretely modeled [17]. Similar to [18], we use ray-casting and (2) to estimate the quality of a vehicle's view. In our approach, we use a geometric ray-sphere transformation to estimate interactions with obstacles and the target efficiently. The set of coordinates that terminate at the robot's maximum observation range  $d_{\text{max}}$  are derived from a set of unit vectors U that emanate from a viewpoint and are defined as follows:

$$\mathbf{X}_s = \{ \boldsymbol{p} + d_{\max} \, \hat{\boldsymbol{u}} \mid \hat{\boldsymbol{u}} \in \mathbf{U} \}.$$
 (5)



Fig. 5. Pictoral depiction of a ray-sphere intersection with coordinate z. The intersection point  $h_{\min}$  is used to update the coordinate  $x_s$ .

We also define the radius  $c_r$  to check each ray for an intersection with the discretized target space **T** or an obstacle coordinate. As illustrated in Fig. 5, each coordinate of interest z (e.g., target or obstacle coordinate) is checked for intersection with each ray defined by the viewpoint and the set **U**. We use a geometric approach to project the line segment between the viewpoint p and the coordinate of interest along the vector  $\hat{u}$  corresponding to the coordinate  $x_s$ . Algorithm 1 gives an overview of the ray-tracing procedure for an example coordinate z.

Algorithm 1 Ray-casting algorithm
Require: $x_s$ ; $\hat{u}$ ; T
1: $y_s = 0$
2: $d_{\text{terminal}} =   \boldsymbol{x}_s - \boldsymbol{p}  _2$
3: $d_{oldsymbol{z}} = (oldsymbol{z} - oldsymbol{p}) \cdot oldsymbol{\hat{u}}$
4: if $d_z > 0$ then
5: $h_{\text{closest}} = p + d_z \hat{u}  \triangleright \text{ Closest point along ray to } z$
6: $d_{\text{center}} =   \boldsymbol{h}_{\text{closest}} - \boldsymbol{z}  _2$
7: <b>if</b> $d_{\text{center}} < c_r$ <b>then</b>
8: $d_{\rm chord} = \sqrt{c_r^2 - d_{\rm center}^2}$
9: $\mathbf{H} = \mathbf{h}_{closest} \pm d_{chord}$
10: $oldsymbol{h}_{\min} = rg\min   oldsymbol{h}_s - oldsymbol{p}  _2$
$h_s \in \mathbf{H}$
11: $a_{\text{intersect}} =   \boldsymbol{n}_{\min} - \boldsymbol{p}  _2$
12: <b>if</b> $d_{\text{intersect}} < d_{\text{terminal}}$ <b>then</b>
13: $d_{\text{terminal}} = d_{\text{intersect}}$
14: $oldsymbol{x}_s \leftarrow oldsymbol{h}_{\min}$
15: <b>if</b> $z \in \mathbf{T}$ then
16: $y_s \leftarrow 1$
17: return $\boldsymbol{x}_s, y_s$

Given that we can use this algorithm to find intersections with target coordinates and obstacles, we update each  $x_s \in \mathbf{X}_s$  using Algorithm 1 and store whether each  $x_s$  is associated with the target, logically denoted by  $y_s \in Y_s$ .

Subsequently, we can determine the values of the variables in (3) and (4) that intersect the target.  $\phi_{i \min}$  and  $\phi_{i \max}$  are the minimum and maximum angles between the normal vector of the frame and the vectors that intersect the target along the *i*<sup>th</sup> dimension. In the same way,  $\ell_{i1}$  and  $\ell_{i2}$  are determined by the distance to the point of intersection between the target and the rays along the *i*<sup>th</sup> dimension. Using this methodology, we can similarly compute the score in (2), but in a discretized manner, increasing the computation speed for each candidate view in the environment.

### C. Target Coverage Estimation using GP Interpolation

The exact coverage for any target in 2D or 3D space is a binary indicator for all coordinates of whether a coordinate of the target has been captured or not. However, this process can be expensive for a vehicle with limited computing capability for any sized target, especially considering that we would need an infinite number of rays to exactly compute the coverage. For this reason, we use sparse ray-casting and Gaussian process interpolation, utilizing induction points based on the target location.

Gaussian process interpolation is a probabilistic framework for interpolation and provides a natural way to quantify uncertainty for a static set of target points [19], [20]. Given a finite and sparse number of sampling points from the ray intersections, we use Gaussian process interpolation to estimate the probability of a target coordinate  $x \in \mathbf{X}$  being observed at non-sampled target points.

Consider an initial GP, as presented in [21]

$$f(\mathbf{X}) \sim \mathcal{GP}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X}'))$$
(6)

characterized by a mean and covariance function:

$$\mu(\mathbf{X}) = \mathbb{E}[f(\mathbf{X})] \tag{7}$$

$$k(\mathbf{X}, \mathbf{X}') = \mathbb{E}[(f(\mathbf{X}) - \mu(\mathbf{X}))(f(\mathbf{X}') - \mu(\mathbf{X}'))] \quad (8)$$

where  $f(\mathbf{X})$  is functionally interpreted in this application as whether a target coordinate  $x \in \mathbf{X}$  has been captured by the inspecting vehicle (i.e., a zero if the coordinate has not been observed and a one if the coordinate has been inspected before). For interpolation of the target space, the widely used radial basis function (RBF) kernel is employed to capture the spatial relationship between two points and likewise interpolate their value using the Gaussian process regression with the kernel equation formulated as:

$$k(\mathbf{X}, \mathbf{X}') = \sigma_f^2 \exp\left(-\frac{||\mathbf{X} - \mathbf{X}'||^2}{2\sigma_l^2}\right).$$
 (9)

where  $\sigma_l \in [0, 1]$  and  $\sigma_f \in [0, 1]$  are two hyperparameters. In this work, we tuned the hyperparameters by minimizing the negative log-likelihood of the training data. In this application, the hyperparameters were tuned using PSO because of its flexibility, given that it could be adapted to a higher speed or higher accuracy depending on the convergence criteria [22]. Additionally, the variable  $\mathbf{X} \in \mathbb{R}^{m \times n_d}$  is a set of msampled target coordinates from the ray-casting discussed in the previous section. We can estimate the target coverage as the difference between the prior target coverage  $\mu(\mathbf{X})$  and the estimated target coverage using the posterior mean,  $\mu^*(\mathbf{X})$ , from a new viewpoint p given the set of sample target points  $\mathbf{X}_s \in \mathbb{R}^{s \times n_d}$ . As such, the computational complexity of the target coverage is significantly reduced by the new equation

$$G(\mathbf{X}, \boldsymbol{p}) = \gamma_d \cdot \gamma_s \cdot \sum_{j=1}^m (\mu_j^*(\mathbf{X}) - \mu_j(\mathbf{X})).$$
(10)

where the subscript on  $\mu$  denotes the element of the mean value for coordinate j and we compute  $\mu^*$  as

$$\mu^*(\mathbf{X}) = k(\mathbf{X}, \mathbf{X}_s) \left[ k(\mathbf{X}_s, \mathbf{X}_s) + \sigma_n^2 I \right]^{-1} Y_s.$$
(11)

The variable  $\sigma_n$  is a constant in this case representing sensor noise and  $Y_s$  is the binary 1D matrix signifying if the sample terminal ray-cast coordinates in  $\mathbf{X}_s$  were in range of a target coordinate. With this belief update, we estimate the probability of additional coverage of the target from a candidate viewpoint p and (2) can be efficiently computed for each candidate viewpoint p. However, computing every candidate view in the environment is intractable at runtime, especially given that (2) may have a non-linear or discontinuous solution space. To overcome this limitation, we incorporate particle swarm optimization to dynamically determine the optimal viewpoint in real-time.

# D. Candidate View Particle Swarm Optimization

As previously stated in Sec. IV-C, PSO is useful for solving a highly non-linear problem and is adaptable to runtime considerations. In our inspection application, a candidate viewpoint is represented as a point in a N-dimensional solution space. A swarm of particles is used to represent potential viewpoints. Each particle,  $p_i(t)$ , is the coordinate of the  $i^{\text{th}}$  candidate view at time t, and its velocity components are represented as  $v_i(t)$ . The coverage G is calculated for each viewpoint and the particles migrate toward the particle with the highest evaluation score.

At each time t, the velocity of a particle is perturbed by the weighted sum of its personal best view and global best view, resulting in an updated velocity and position. To update the particle, we use the following equations:

$$\boldsymbol{v}_i(t+1) = w \boldsymbol{v}_i(t) + r_1 c_1 (\boldsymbol{b}_i(t) - \boldsymbol{p}_i(t)) \qquad (12)$$
$$+ r_2 c_2 (\boldsymbol{g}(t) - \boldsymbol{p}_i(t))$$

$$p_i(t+1) = p_i(t) + v_i(t+1)$$
 (13)

where  $b_i(t)$  and g(t) are the global best position evaluated by the particles and the global best position of all particles, respectively. The acceleration coefficients  $c_1$  and  $c_2$  are nonnegative constants, which weigh the influence of the personal and global bests in the search process. The inertia weight w balances the local and global exploration of the particles in the search space [23]. To avoid divergence, the value of w must be between 0 and less than 1. The coefficients  $c_1$ and  $c_2$  can be adjusted to influence the particles to explore the area around their local optima (when  $c_1 > c_2$ ) or the area around the global optima (when  $c_2 > c_1$ ). This can be changed during the heuristic process to promote exploration or exploitation. Variables  $r_1$  and  $r_2$  are random numbers between 0 and 1 and are used to perturb the particles to encourage exploration.

Fig. 7 shows the result of our approach in 3D in which the vehicle detects an obstacle that blocks the target and propagates particles given the updated occupancy map. The vehicle moves toward the best view, as defined by the best global position of all particles. As the vehicle moves, the particles iteratively migrate in search of the best viewpoint. Once the vehicle reaches the best viewpoint location, it will capture a photo and update the observed portions of the target as shown in Fig. 7(b). This allows the vehicle to move to unobserved portions of the target in future iterations. In this instance, the best view of the target was shifted due to the obstacle. Since the raycasting is uniform from the vehicles' position, the right side of the target was estimated as unseen rather than fully reflecting the ground truth outcome. We suggest that this is a beneficial outcome, as the vehicle was able to travel to the estimated "unseen" part of the target to obtain an undistorted image of the other side rather than the small left-hand portion.



Fig. 6. An example using GP interpolation to estimate the visibility of a target in a two-dimensional space. The color bar to the right of (a) displays the ground truth target coverage, while (b) illustrates the estimation of the coverage using GP interpolation.



Fig. 7. An example of using GP to estimate the visibility of the target in 3-dimensional space. (a) presents the settings, (b) shows the estimated GP result from the ray hits. (c) provides the ground truth for visibility.

### E. Comparison Discussion

We note that this is a novel approach for photographing and inspecting targets, minimizing the number of photos to capture and evaluating the quality of the images. Therefore, other methods can incorporate these scoring metrics as part of their planning approach. We demonstrate an example of such modification, where we compare a sampling-based frontier approach from [24] applied to a target space in Fig. 8. Fig. 8(a) shows that the typical sampling-based frontier approach will value efficient exploration and consider target areas explored. The frontier approach captures a new image each time a previously unseen portion of the target is uncovered. It takes a total of 25 photos, each of which fails to match the quality metrics defined in our work. In contrast, the proposed approach shown in Fig. 8(b) only takes 2 inspection photos that maximize the designed evaluation metrics.



Fig. 8. Comparison of sampling-based frontier approach (a) with the proposed NBV-based approach (b).

### V. SIMULATIONS

To demonstrate the robustness of the proposed method, we conducted extensive simulations using different vehicles equipped with sensors of varying capabilities in diverse environmental contexts. In these simulations, the vehicle operates in a partially-known environment defined by operational boundaries, target position, and dimensions. As stated in Sec. III, we assume that the vehicle is equipped with two primary sensor types: 1) a depth sensor for obstacle detection (e.g., RGB-D camera, Lidar, etc.), and 2) an optical camera used for photography. We ran all tests in MATLAB on a Lenovo ThinkPad P14s laptop with a 2.80 GHz Intel 4-Core i7 processor and 32 GB RAM.

In the 2D simulation, the vehicle operates in a  $100m \times 100m$  grid world with a resolution of 1m and uses a hybrid A\* for path planning, benefiting from its ability to rapidly compute optimal paths in 2D environments while accounting for vehicle dynamics. The objective is to inspect the four sides of a building measuring  $30m \times 15m$ . Various shapes of unknown obstacles are placed in front of three sides of the building. Both the depth sensor and the camera have a field of view (FOV) of  $\Phi = [-\frac{\pi}{4}, \frac{\pi}{4}]$ . The range of the RGB-D sensor is  $r_{max} = 25m$ . We set  $\beta = 0.8$  and the utility functions are designed in the style of logistic functions:

$$U(x, \mathbf{q}) = q_1 + \frac{q_2}{1 + e^{q_3(x+q_4)}}, \quad \mathbf{q} = [q_1, ..., q_4]$$
(14)

We choose logistic functions because their inherent property of rapid decline in value allows for the effective differentiation between preferred values and those less desirable. For the utility function  $U_d$ , a more substantial penalty should be applied as x approaches 1, indicating increased perspective distortion. Therefore, we select  $q_d = [0.3, 0.7, 20, -0.75]$ . Regarding  $U_s$ , we construct a piece-wise function with  $q_s =$ [0, 1, -20, -0.5] for  $x \in [0, \beta]$  and  $\mathbf{q}_s = [0, 1, 30, -1]$ for  $x \in (\beta, 1]$  to ensure the utility's peak aligns with the desired proportion  $\beta$ .  $U_d$  diminishes as x diverges from  $\beta$ , encouraging  $\beta$  proportion of the target within the frame while preventing the view from intersecting with the frame edges [1]. Users can also alter these functions or design functions in similar styles to fit their applications. The utility functions are not dependent on the target or environment and were used in all simulations and experiments.

We use PSO to search for the optimal viewpoint by uniformly distributing 20 particles across the environment and iteratively refining the search. Fig. 9 shows the topdown view of some keyframes during the simulation. The heatmap on the second row is generated by computing the expected target coverage and image quality for every viewpoint in the environment using (2). As illustrated in Fig. 9, the vehicle takes pictures from the best viewpoint found by the PSO on each side of the building and takes several pictures from different points of view for the same target when necessary. The L2 norm of the error between the global optimal and the results obtained through PSO is  $0.8538 \pm 0.6816m$ . Considering the dimension of the target, this error is neglectable. It's noteworthy that this error could be mitigated by either adding more particles or increasing the number of optimization iterations. GP estimates of the inspected portions are shown as colored bars on the target. The evaluation performance for this 2D example occurs at a rate of 1.79ms per candidate viewpoint.

A 3D simulation result is shown in Fig. 10 in which a  $10m \times 1m \times 10m$  target is positioned in a  $30m \times 30m \times 40m$  space with an obstacle placed in front of it. In this case, we employ RRT\* for its speed and efficiency in computing a path in a complex 3D environment. The RGB-D sensor has identical horizontal and vertical FOV. As indicated in Fig. 10, the aerial vehicle routes to the optimal viewpoint from where it can see the majority of the target and avoids the view from being occluded by the obstacle. The corresponding heatmaps orthogonal at the viewpoint are provided to show the optimality of the viewpoint. The evaluation for the 3D example occurs at a rate of 5.48ms per candidate viewpoint.



Fig. 9. A top-down perspective of a 2D simulation featuring a quadrotor assigned to inspect four sides of a building. The frames are chronologically arranged from left to right. The first row captures 1) the quadrotor states, 2) exploration progress, and 3) the inspected portion of the target. The heatmaps in the second row show the ground truth information gain at all viewpoints in the environment.



Fig. 10. Heatmap showing the results of the equations in the x-y, x-z, and y-z planes, sliced at the optimal locations (denoted by a black star). As shown, the vehicle is at the optimal point and can see the whole target.

### VI. EXPERIMENTS

To validate the effectiveness and versatility of our proposed approach, extensive real and virtual experiments were conducted with various autonomous vehicles. We performed 2D experiments on different robots in similar environmental settings with and without the presence of obstacles. The experiments were carried out indoors using a VICON motion capture system to localize the robots. The robots used for these experiments are: a ROSbot skid steering UGV equipped with an RPLIDAR and a camera, and a DJI Tello UAV operating in the x-y plane while maintaining a fixed altitude. As in the simulation, we set  $\beta$  at 0.8.

First, to test the effectiveness of our approach, the Tello was started from the corner of the room and tasked to inspect an object in a free space. From the left to the right in Fig. 11, we demonstrate: 1) the bird view of Tello taking a picture, 2) the GP result shows the entire target is inspected, 3) the heatmap shows the ground truth information gain each viewpoint in the environment (with the UAV at the optimal position), and 4) the target lays in the center and takes up the desired 80% of the horizontal FOV in the final picture.

Fig. 12 shows the result for a case with obstacles blocking the direct view of the facade of a target object. Thanks to our approach, the UAV realizes that it needs to take two pictures from each side of the T-shape obstacle to cover the entire front surface of the object and it only needs to take one perspective shot to look over the obstacle and inspect the entire side surface.

Fig. 13 shows results for a setting with a slash-shaped obstacle. This time the ROSbot UGV locates the best vantage point at which the obstacle appears as a thin line.

To further reinforce these results, we performed virtual experiments using a simulated RotorS Firefly equipped with a stereo camera. This experiment ran on the same hardware



Fig. 11. Without obstacles, the proposed approach enables the UAV to capture the entire target from the optimal inspecting point.



Fig. 12. A 2D experiment is shown that is similar to the 2D simulation. The heatmap shows that the quadrotor captures images from the most advantageous perspectives. The photographs obtained are also displayed.



Fig. 13. An experiment inspecting an object with a slash-shape obstacle. The ground vehicle preserves almost the entire target with minimal distortion. The obstacle appears as a thin line.

in Sec. V in C++. The evaluation of each candidate viewpoint occurs in 2.39ms. Fig. 14 shows the results of an example virtual experiment. As shown in Fig. 14, the vehicle detects the tree in front of the house using the stereo camera and stores the readings in an OctoMap [25]. The vehicle uses particle swarm optimization to maximize the equation in (2) where the result is a viewpoint that maximizes the information gain while minimizing distortion. The picture in Fig. 14(b) shows the resulting image from this viewpoint. For this experiment, we set  $\beta$  to 0.8 and use RRT\* to route the



Fig. 14. An example virtual experiment where a tree obstructs the view of the target. (a) and (b) show the initial and final view. (c) and (d) show the trajectory followed to the best viewpoint in gazebo and rviz.

vehicle around obstacles. The Gazebo snapshot in Fig. 14(c) shows the planned trajectory of the UAV. The Rviz snapshot in Fig. 14(d) shows the resulting plan given the observed obstacles, where the red-shaded region represents the target.

### VII. DISCUSSION AND CONCLUSION

In this work, we have presented a novel framework for an autonomous vehicle to take a quality image of a target in a partially-known environment. The approach includes utility functions to define the quality of a viewpoint as well as a method to estimate the information gain of a viewpoint at runtime. The extensive simulations and results of the experiments show the validity, applicability, and generality of the proposed method.

Moving forward we are interested in investigating the following directions: 1) incorporate our metrics into existing NBV planners to optimally route to viewpoints; 2) extend the framework to consider dynamic obstacles; 3) identify areas of interest to inspect after the entire target has been viewed such as structural inconsistencies and areas of concern; 4) implement this approach with only a monocular camera, utilizing machine learning to recognize depth information of a target; 5) conduct modeling and testing of this framework with multiple robots.

### VIII. ACKNOWLEDGEMENTS

This work is based on research sponsored by the CoStar Group. We would like to thank Khajamoinuddin Syed for the useful discussion about the research application.

### REFERENCES

- M. Zabarauskas and S. Cameron, "Luke: An autonomous robot photographer," in 2014 ieee international conference on robotics and automation (icra). IEEE, 2014, pp. 1809–1815.
- [2] B. Sutter, A. Lelevé, M. T. Pham, O. Gouin, N. Jupille, M. Kuhn, P. Lulé, P. Michaud, and P. Rémy, "A semi-autonomous mobile robot for bridge inspection," *Automation in Construction*, vol. 91, pp. 111– 119, 2018.
- [3] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Q. Li, and I. Rekleitis, "Underwater exploration and mapping," in 2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), 2022, pp. 1–7.

- [4] Z. Byers, M. Dixon, K. Goodier, C. M. Grimm, and W. D. Smart, "An autonomous robot photographer," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 3. IEEE, 2003, pp. 2636–2641.
- [5] R. Montero, J. G. Victores, S. Martinez, A. Jardón, and C. Balaguer, "Past, present and future of robotic tunnel inspection," *Automation in Construction*, vol. 59, pp. 99–112, 2015.
- [6] C. Connolly, "The determination of next best views," in *Proceedings*. 1985 IEEE international conference on robotics and automation, vol. 2. IEEE, 1985, pp. 432–435.
- [7] M. Naazare, F. G. Rosas, and D. Schulz, "Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3779– 3786, 2022.
- [8] Y. Han, I. H. Zhan, W. Zhao, and Y.-J. Liu, "A double branch next-best-view network and novel robot system for active object reconstruction," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 7306–7312.
  [9] H. Dhami, V. D. Sharma, and P. Tokekar, "Map-nbv: Multi-agent
- [9] H. Dhami, V. D. Sharma, and P. Tokekar, "Map-nbv: Multi-agent prediction-guided next-best-view planning for active 3d object reconstruction," *arXiv preprint arXiv:2307.04004*, 2023.
  [10] X. Zeng, T. Zaenker, and M. Bennewitz, "Deep reinforcement learning
- [10] X. Zeng, T. Zaenker, and M. Bennewitz, "Deep reinforcement learning for next-best-view planning in agricultural applications," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 2323–2329.
- [11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in 2016 IEEE international conference on robotics and automation (ICRA). IEEE, 2016, pp. 1462–1468.
- [12] R. Almadhoun, A. Abduldayem, T. Taha, L. Seneviratne, and Y. Zweiri, "Guided next best view for 3d reconstruction of large complex structures," *Remote Sensing*, vol. 11, no. 20, p. 2440, 2019.
- [13] R. Monica, J. Aleotti, and D. Piccinini, "Humanoid robot next best view planning under occlusions using body movement primitives," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 2493–2500.
- [14] I. Lluvia, E. Lazkano, and A. Ansuategi, "Active mapping and robot exploration: A survey," *Sensors*, vol. 21, no. 7, p. 2445, 2021.
- [15] D. Friedman and Y. A. Feldman, "Automated cinematic reasoning about camera behavior," *Expert Systems with Applications*, vol. 30, no. 4, pp. 694–704, 2006.
- [16] A. Asgharivaskasi and N. Atanasov, "Active bayesian multi-class mapping from range and semantic segmentation observations," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 1–7.
- [17] J. I. Vásquez-Gómez, E. Löpez-Damian, and L. E. Sucar, "View planning for 3d object reconstruction," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009, pp. 4015– 4020.
- [18] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "Hierarchical ray tracing for fast volumetric next-best-view planning," in 2013 International conference on computer and robot vision. IEEE, 2013, pp. 181–187.
- [19] L. Yel and N. Bezzo, "Gp-based runtime planning, learning, and recovery for safe uav operations under unforeseen disturbances," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 2173–2180.
- [20] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communicationefficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.
  [21] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine*
- [21] C. K. Williams and C. E. Rasmussen, Gaussian processes for machine learning. MIT press Cambridge, MA, 2006, vol. 2, no. 3.
- [22] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, "Selecting hyper-parameters of gaussian process regression based on non-inertial particle swarm optimization in internet of things," *IEEE Access*, vol. 7, pp. 59504–59513, 2019.
- [23] Y. Zhang, D.-w. Gong, and J.-h. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.
- [24] V. M. Respall, D. Devitt, R. Fedorenko, and A. Klimchik, "Fast sampling-based next-best-view exploration algorithm for a may," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 89–95.
- [25] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.