Neural Observation Field Guided Hybrid Optimization of Camera Placement

Yihan Cao[1], Jiazhao Zhang[2], Zhinan Yu[1], and Kai Xu[1,†]

***Abstract*— Camera placement is crutial in multi-camera systems such as virtual reality, autonomous driving, and high-quality reconstruction. The camera placement challenge lies in the nonlinear nature of high-dimensional parameters and the unavailability of gradients for target functions like coverage and visibility. Consequently, most existing methods tackle this challenge by leveraging non-gradient-based optimization methods. In this work, we present a hybrid camera placement optimization approach that incorporates both gradient-based and non-gradient-based optimization methods. This design allows our method to enjoy the advantages of smooth optimization convergence and robustness from gradient-based and non-gradient-based optimization, respectively. To bridge the two disparate optimization methods, we propose a neural observation field, which implicitly encodes the coverage and observation quality. The neural observation field provides the measurements of the camera observations and corresponding gradients without the assumption of target scenes, making our method applicable to diverse scenarios, including 2D planar shapes, 3D objects, and room-scale 3D scenes. Extensive experiments on diverse datasets demonstrate that our method achieves state-of-the-art performance, while requiring only a fraction (8x less) of the typical computation time. Furthermore, we conducted a real-world experiment using a custom-built capture system, confirming the resilience of our approach to real-world environmental noise. We provide code and data at: https://github.com/yhanCao/MultiviewOpt.**

I. INTRODUCTION

Camera placement is a long-standing and widely applicable problem [1], [2] across various domains such as motion tracking [3], [4], surveillance systems [5], and robotics [6]. Through analysis of 3D spatial priors, camera placement methods optimize placement of multi-cameras, to maximize visibility metrics such as coverage. However, the camera placement problem faces two primary challenges: firstly, a highly non-linear optimization landscape due to the high dimensionality of multi-camera parameters; secondly, the process of calculating visibility lacks differentiability, rendering the gradient of visibility unattainable.

To tackle this challenge, most existing methods [5], [7], [8], [9] leverage non-gradient optimization methods with explicit scene representation, such as point clouds or voxels. These methods typically adopt heuristic or greedy search algorithms, eliminating the need for gradients. However, they often suffer from issues related to convergence speed and precision. Additionally, explicit scene representations are predefined for specific scenarios, such as 2D planar shapes or 3D objects, leading to significant performance drops under different scenarios.

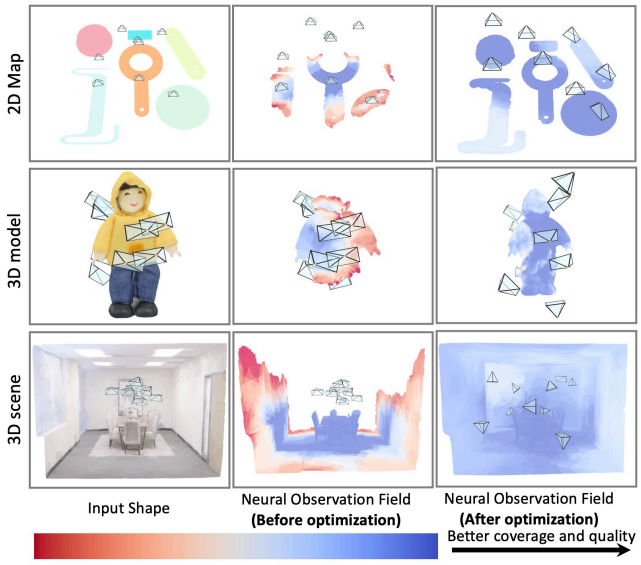


Fig. 1: We introduce a hybrid optimization method based on the neural observation field for camera placement estimation. The target objects are represented by neural observation fields, which are compatible with any type of objects.

In this work, we propose a novel hybrid optimization approach for camera placement optimization, incorporating both gradient-based and non-gradient-based optimization methods. Fig.1 illustrates the hybrid optimization results based on our proposed neural observation field.

For gradient-based optimization, we propose a neural observation field designed to encode scene observation, facilitating gradient calculation for maximizing coverage and observation quality. This neural observation field, represented by an implicit neural field, encodes the joint configuration of scene geometry with multi-camera placement, thereby implicitly capturing multi-view observability. Throughout the optimization process, the neural observation field undergoes online updates following each new camera placement iteration. Leveraging the differentiability of the neural observation field, we compute gradients through gradient backward propagation and subsequently update camera placements. This approach ensures a smooth and rapid convergence of camera placement optimization.

For non-gradient-based optimization, we leverage an elite selection algorithm that retains cameras with high visibility

This work was supported in part by the NSFC (62325211, 62132021) and the Major Program of Xiangjiang Laboratory (23XJ01009).

[1] National University of Defense Technology.
[2] CFCS, School of Computer Science, Peking University.
[†] Corresponding Author.

while resampling those with poor visibility. The cameras with poor visibility are relocated to areas requiring more observation globally. This strategy reduces the risk of getting stuck in local optima and enhances the robustness of our method in complex scenarios, such as regions with large gaps or non-convex geometries. Unlike approaches such as simulated annealing [10] or evolutionary strategies [11], which involve resampling the placement of all cameras, our method focuses solely on resampling cameras with poor visibility, as identified by the convergent results of gradient-based optimization. This approach is more efficient since cameras with good placement exhibit smaller gradients, which remain relatively consistent during gradient-based optimizations.

We conduct extensive experiments on diverse datasets, including 2D planar shapes, 3D objects [12], and room-scale 3D scenes [13]. We evaluate our method on both coverage and observation quality. The results demonstrate that our approach outperforms existing solutions while requiring only a fraction (8x less) of the typical computational time. Furthermore, we develop a custom capture system to evaluate our method's resilience to real-world environmental noise. We find that our method demonstrates robustness and outperforms existing methods across a wide range of objects.

In summary, our main contributions include:

- A hybrid camera placement optimization method that cooperatively incorporates both gradient-based and non-gradient-based optimization methods.
- A neural observation field that implicitly encodes the geometry priors and observation of indoor scenes.
- A custom-built capture system powered by our camera placement optimization method, demonstrating robustness and well-visibility in the real-world environments.

II. RELATED WORKS

A. Camera Placement Application.

Multi-camera systems are extensively employed for capturing and synthesizing realistic 3D content in both research and real-world development scenarios [1]. Visual sensors, known for their low cost, lightweight, and image capture capabilities in various domains. Multi-view applications have emerged in fields such as industrial inspection [14], surveillance [6], [5], motion capture [4], and navigation [15], [16], overcoming the limitations of single-camera vision.

B. Camera Placement Optimization Methods.

Multi-view optimization is a well-established NP-hard problem, making it impractical and time-consuming to find the optimal solution. The focus of this study is to find a sub-optimal solution within a reasonable time frame. Various approaches have been employed for this task [17], [18]. Greedy algorithms, as utilized in [19], iteratively consider all visibility factors until achieving high-quality results. The computation time of the greedy algorithm escalates with scenario scope. Considering time overhead, [20], [7], [10] optimize camera layouts using heuristic algorithms. [21] employs Integer programming algorithm to optimize visibility, single-best coverage quality, and cumulative quality. [22] utilizes Particle Swarm Optimization (PSO) to minimize measurement error pixel quantization and measurement based on the environment model. However, the PSO algorithm's high computation time is noted in [9]. While considering reconstruction, camera placement problem escalates from a single-coverage problem to a multi-coverage problem. The placement of cameras also significantly influences the quality of the reconstructed 3D content. [4] uses Simulated Annealing Algorithm (SA) to optimize camera placements by estimating the three-dimensional positions of markers through triangulation from multiple cameras. For 3D reconstruction, bundle-based methods in [23], [24], [25] utilize photogrammetric camera networks or expert systems to obtain automated camera placement, considering only the reconstruction quality.

The heuristic function for these tasks primarily revolves around the coverage rate of the entire scenario. Additionally, [8] minimizes the coverage optimality gap, defined as the squared error between the desired and achieved coverage. This metric serves as our coverage metric for assessing multi-camera coverage.

Greedy-based or heuristic algorithms overlook the prior information regarding the shape of the scenario, resulting in time wastage in irrelevant areas. Moreover, the spatial discretization of solutions can lead to sub-optimal results, as it depends on the scale of spatial division. Our method incorporates both gradient-based and non-gradient-based optimization methods by taking advantage of scenario geometry prior to mitigate these challenges.

C. Scene Representation for optimization.

Scene representation in the camera placement problem involves unstructured point clouds [22], volumes [8], or polygonal meshes [9]. However, model represented by points or meshes may degrade the result due to irregular distributions of points or faces, which fail to adequately describe the coverage of model space, influenced by density [26]. Voxelized scene representation, while effective in some cases, cannot facilitate gradient backpropagation for gradient-based optimization. Recently, implicit representation of shapes has made significant advancements. Implicit model field methods typically aim to learn a function that maps spatial locations to feature representations. [27] proposes neural sparse voxel fields as a novel neural scene representation for fast and high-quality free-viewpoint rendering. In [28], a constructed voxel field representation is utilized for cross-modality 3D object detection. Inspired by these works, we encode the scenario geometry prior to obtain the visibility of scenes.

III. METHOD

A. Problem Statement and Method Overview

Given a target object $\mathcal{S} = \{\mathbf{s}_j, \mathbf{n}_j | \mathbf{s} \in \mathbb{R}^3, \mathbf{n} \in \mathbb{R}^3\}_{j=0:n}$ represented by a point cloud $\{\mathbf{s}\}_{0:n}$ and corresponding

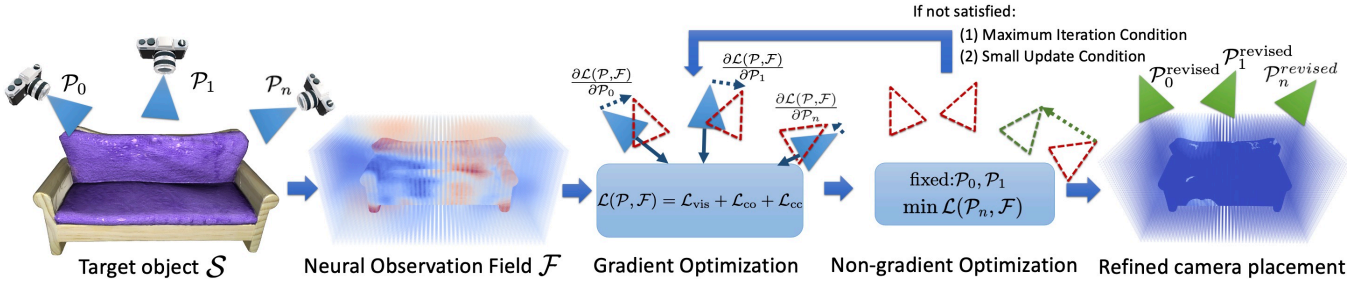


Fig. 2: Method overview. Our method takes target object $\mathcal{S}$ and initial camera placement $\{\mathcal{P}_0, \mathcal{P}_1, ..., \mathcal{P}_n\}$ as inputs to construct neural observation field $\mathcal{F}$. We then utilize the non-gradient-based optimization techniques along with gradient-based optimization methods for camera placement refinement. Throughout this optimization process, the neural observation field is continually updated (refer to section 1), until the termination criteria are met.

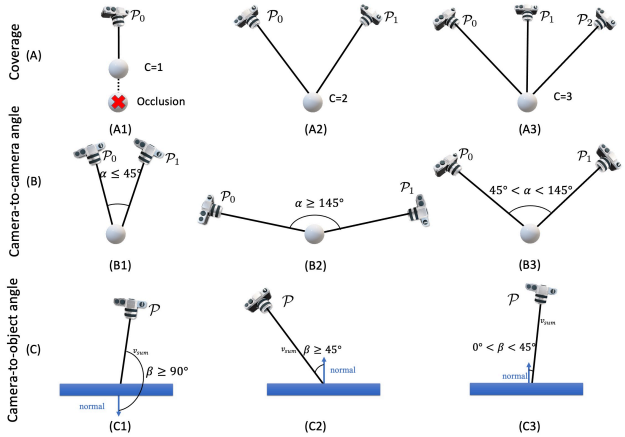


Fig. 3: Illustration of observation attribute $\mathbf{o}$ elements: Coverage c in row (A), Camera-to-camera angle in row (B) and Camera-to-object angle in row (C). Only the third column satisfies our visibility condition.

normal $\{\mathbf{n}\}_{0:n}$ (where n indicates the number of points), the camera placement optimization method aims to maximize the visibility of the target scene, such as coverage or observation quality. The camera placement $\mathcal{P} = \langle \mathbf{p}_i, \mathbf{r}_i \rangle_{i=1:k}$ involves their positions $\{\mathbf{p}_i | \mathbf{p} \in \mathbb{R}^3\}_{i=1:k}$ and orientations $\{\mathbf{r}_i | \mathbf{r} \in SO(3)\}_{i=1:k}$ (where k indicates the number of cameras).

Our method leverages a hybrid optimization method based on the neural observation field. The neural observation field $\mathcal{F} : (\mathcal{S}, \mathcal{P}) \rightarrow (c, \phi^{cc}, \phi^{co})$ implicitly encodes scene priors including *coverage* $(c_i)_{i=1:n}$ (the number of visible cameras), average *camera-to-camera* viewing angle $(\phi_i^{cc})_{i=1:n}$ and average *camera-to-object* viewing angle $(\phi_i^{co})_{i=1:n}$ (between camera ray and target normal vector)(Section III-B). Then a hybrid optimization method $\mathcal{H} : (\mathcal{P}_t, \mathcal{S}, \mathcal{F}_t) \rightarrow (\mathcal{P}_{t+1}, \mathcal{F}_{t+1})$ (Section III-C) is conducted by iteratively performing the gradient-based optimization $\mathcal{H}_{grad}(\mathcal{P}, \mathcal{F})$ and non-gradient-based optimization $\mathcal{H}_{non\_grad}(\mathcal{P}, \mathcal{F})$ to achieve a good trade-off between exploitation and exploration in camera placement optimization. An overview visualization of our pipeline is presented in Figure 2.

B. Neural Observation Field

To obtain the visibility of scenes, we divide the target object $\mathcal{S}$ into voxels $\mathcal{V}$. Given the current camera poses, we can determine the voxels that are visible in a single view. After acquiring the observation attributes $o_{\mathcal{V}}$ of voxels $\mathcal{V}$, we encode the joint configuration of scene geometry and voxels with observation attributes $o_{\mathcal{V}}$ to derive the neural observation field $\mathcal{F}_t$ of the target object $\mathcal{S}$ and current camera placement $\mathcal{P}_t$.

a) Observation Attributes: We define the observation attributes, consisting of three elements: Coverage c defined by the coverage relationship E, Camera-to-camera angle ϕ^{cc}, and Camera-to-object angle ϕ^{co} to represent the visibility of voxels. We consider the field of view (FOV), image blur, and occlusion to determine visibility. Voxels within the camera's frustum that are not obscured by other voxels in this view are considered visible to that camera.

The coverage relationship $E(i, j)$ between a voxel $\mathcal{V}_j$ and a camera C_i is represented as a binary number, where $E(i, j) = 1$ if $\mathcal{V}j$ is visible in C_i, otherwise $E(i, j) = 0$. For a given coverage threshold K, the coverage condition that voxel j still needs is expressed as :

$$c(j) = K - \sum_{i=0}^{k} E(i, j). \tag{1}$$

When considering the context of multi-view reconstruction, it is imperative to ensure the accuracy of the computed target 3D location. This accuracy relies on the error propagation characteristics of the reconstruction linear solver. In cases where the rays emanating from the view center towards the target point are either parallel or nearly parallel, a valid solution cannot be obtained [4]. Based on the fulfillment of coverage condition, different target points may have varying numbers of rays intersecting them, necessary to adopt a scale-invariant representation to assess the quality of coverage. To quantify the observation quality of voxel j, we utilize two metrics: Camera-to-camera angle ϕ_j^{cc}, an internal variability formulated as:

$$\phi_j^{cc} = \left| \frac{\pi}{2} - \frac{1}{C_{|A_j|}^2} \sum_{\substack{a_1, a_2 \in A_j \\ a_1 \neq a_2}} \alpha(a_1, a_2) \right|, \tag{2}$$

and Camera-to-object angle ϕ_j^{co}, an external similarity formulated as :

$$\phi_j^{co} = 1 - \frac{\mathbf{n}_j \cdot \sum_{a \in A_j} a}{\|\mathbf{n}_j\| \cdot \left\| \sum_{a \in A_j} a \right\|}, \tag{3}$$

where A_j is the set of vectors from the center of voxel j to the cameras that observe voxel j, α is the angle between two vectors, and $C^2_{|A_j|}$ is a combinatorial term representing the selection of all pairs of vectors from A_j. The observation attributes consist of three fundamental elements: $\mathbf{o} = [c, \phi^{cc}, \phi^{co}]$ illustrated in Figure 3 .

b) Neural Observation Field: Each voxel $\mathcal{V}_j$ is characterized by its center position $\mathbf{s}_j$, normal $\mathbf{n}_j$ and observation attributes $[c_j, \phi^{cc}_j, \phi^{co}_j]$. The neural observation field $\mathcal{F}$ operates by implicitly encoding scene prior perception and offers a continuous and efficient observation query mechanism for optimization, corresponding to the function *LeanNeOF* in Algorithm. 1.With a fixed number of cameras, surface points observed by each camera are expected to be parts of objects that are less observed in current camera placement.

We employ the Scaled Dot-Product Attention function, which enables the model to focus more on the parts that are relevant to the current or other contextual information [29]. This function is denoted as $\mathcal{F}$, which aggregates target object surface information to obtain query point attributes and optimization directions to optimize corresponding cameras. Common methods for aggregating information for 3D point clouds or voxels are trilinear interpolation [30] or KNN algorithm [31]. However, these methods can only aggregate information from nearby small areas, potentially slowing down camera optimization or getting stuck in locally optimal solutions. According to the concentration mechanism of Attention [29], we can adaptly learn the appropriate weights of all known voxels with attributes via gradient backpropagation.

We compute the relative position and normal of voxels on $\mathcal{S}$ centered at $\mathbf{S}_i$ and then process through an MLP layer(using ReLU activation and 32 channels) to serve as input X of queries and keys. Subsequently, X is separately multiplied with query and key weight matrices denoted as

$$Q = XW^Q, \qquad K = XW^K. \tag{4}$$

W^Q, W^K are the weight matrices for queries and keys, respectively. The observation attribute of target object $\mathcal{S}$ is calculated as:

$$\mathbf{o}_{\mathcal{S}} = softmax(\frac{QK^T}{\sqrt{d_k}})\ \mathbf{o}_{\mathcal{V}}, \tag{5}$$

where d_k is the dimension of the keys. Calculating by Eq. 5, target object $\mathcal{S}$ have equivalent attributes $\mathbf{o}_{\mathcal{S}}$. According to Eq.2, 3, Attribute $\mathbf{o}_{\mathcal{S}} = [c, \phi^{cc}\phi^{co}]$ has a maximum value $sup = [K, \pi/2, 1]$. The sum attributes of all points in all views have a maximum value as well :

$$\sum_{i=1}^{k}\sum_{j=1}^{|\mathcal{S}_i|}\mathbf{o}_{\mathcal{S}_{ij}} \leq \sum_{i=1}^{k}\sum_{j=1}^{n} sup, \tag{6}$$

where $|\mathcal{S}_i|$ is the number of points in view i and $\mathcal{S}_{ij}$ is the jth point of points in view i. After generating the Neural Observation Field, we optimize the camera placement by a hybrid optimization method.

Algorithm 1: Hybrid camera placement optimization $\mathcal{H}$

Input : Target object $\mathcal{S}$
Output: Optimized camera placement $\mathcal{P}$

$\mathcal{P}_0 \leftarrow$ *Initialize*($\mathcal{S}$);
$c_0, \phi^{co}_0, \phi^{cc}_0 \leftarrow$ *ShapeAnalyze*($\mathcal{S}$, $\mathcal{P}_0$);
$\mathcal{F}_0 \leftarrow$ *LeanNeOF*($c_0, \phi^{co}_0, \phi^{cc}_0, \mathcal{S}$);
$\mathcal{L}_0 \leftarrow \infty$; $t \leftarrow 0$; `// Initialize the NeOF`
ex
repeat
  $\mathcal{P}_{t+1}, \mathcal{L}_{t+1} \leftarrow \mathcal{H}_{grad}(\mathcal{P}_t, \mathcal{F}_t)$;
  $c_{t+1}, \phi^{co}_{t+1}, \phi^{cc}_{t+1} \leftarrow$ *ShapeAnalyze*($\mathcal{S}$, $\mathcal{P}_{t+1}$);
  $\mathcal{F}_{t+1} \leftarrow$ *LeanNeOF*(c_{t+1}, ϕ^{co}_{t+1}, ϕ^{cc}_{t+1}, $\mathcal{S}$, $\mathcal{F}_t$);
    `// Fine-tune the NeOF`
  if *CheckLossUpdate*($\mathcal{L}_t, \mathcal{L}_{t+1}$)$< 1e^{-4}$ **then**
    $\mathcal{P}_{t+1} \leftarrow \mathcal{H}_{non\_grad}(\mathcal{P}_{t+1}, \mathcal{F}_{t+1})$;
    $c_{t+1}, \phi^{co}_{t+1}, \phi^{cc}_{t+1} \leftarrow$ *ShapeAnalyze*($\mathcal{S}$, $\mathcal{P}_{t+1}$);
    $\mathcal{F}_{t+1} \leftarrow$ *LeanNeOF*($c_{t+1}, \phi^{co}_{t+1}, \phi^{cc}_{t+1}, \mathcal{S}, \mathcal{F}_{t+1}$);
      `// Extensively fine-tune the NeOF`
  $\mathcal{L}_t \leftarrow \mathcal{L}_{t+1}$; $t \leftarrow t+1$;
until *CheckStepUpdate*($\mathcal{P}_t$, $\mathcal{P}_{t-1}$) $< 10^{-4}$;

C. *Hybrid Placement Optimization method*

Utilizing the differentiable and efficient scene priors query facilitated by the neural observation field, our approach strategically employs both gradient-based optimization and non-gradient-based optimization to strike a favorable balance between exploitation and exploration. Gradient-based optimization enables direct access to gradients derived from the neural observation field, facilitating fine-grained optimization. Nevertheless, it is susceptible to being trapped in local optima. To mitigate this risk and escape local optima, non-gradient-based optimization comes into play. Non-gradient-based optimization identifies and stabilizes camera poses associated with superior coverage and observation quality, subsequently reevaluating sub-optimal camera poses by leveraging analysis of scene priors and convergent results of gradient-based optimization method. For a more comprehensive understanding of the methodology, the details can be found in Algorithm. 1.

Gradient-based camera placement optimization. Gradient optimization is widely used to have fine-grained updating steps for high-quality optimization. With the differentiable neural observation field $\mathcal{F}$, our method can obtain the gradient via a self-defined loss. Based on Eq. 5, 6, maximizing the coverage and observation quality is equivalent to minimize

$$[\mathcal{L}_{vis}, \mathcal{L}_{cc}, \mathcal{L}_{co}] = sup - \frac{\sum_{i=1}^{k}\sum_{j=1}^{|\mathcal{S}_i|}\mathbf{o}_{\mathcal{S}_{ij}}}{k \cdot n}, \tag{7}$$

where $[\mathcal{L}_{vis}, \mathcal{L}_{cc}, \mathcal{L}_{co}]$ are three different metrics, including (1) Coverage loss $\mathcal{L}_{vis}$ guiding the camera to maximize its observation of objects, (2) Camera-to-camera viewing angle loss $\mathcal{L}_{cc}$, to enable triangular perception, and (3) Camera-to-object viewing angle loss $\mathcal{L}_{co}$, to enable cameras to face directly to the object.

To increase the coverage and observation quality, we use $\mathcal{H}_{grad}(\mathcal{P}, \mathcal{F})$ to optimize the camera placement $\mathcal{P} =$

$\langle \mathbf{p}_i, \mathbf{r}_i \rangle_{i=1:k}$ through a combination loss of three metrics:

$$\mathcal{L}(\mathcal{P}, \mathcal{F}) = w_{vis}\mathcal{L}_{vis} + w_{cc}\mathcal{L}_{cc} + w_{co}\mathcal{L}_{co}, \tag{8}$$

where the w_{vis}, w_{cc} and w_{co} are the weights of each cost term. We empirically set the weights as $w_{vis} = 0.4$, $w_{cc} = 0.3$ and $w_{co} = 0.3$, and the weights can be flexibly adjusted for specific requirements, like maximizing the visibility ($w_{vis} = 1.0$, $w_{cc} = 0.0$ and $w_{co} = 0.0$).

Camera poses $\mathcal{P} = \langle \mathbf{p}, \mathbf{r} \rangle$ are optimized using the gradient of $\mathcal{L}(\mathcal{P}, \mathcal{F})$ as the optimization parameters. The camera transforms the point cloud observed in its view to the Neural Observation Field coordinate system using its camera parameters and calculates the loss. Due to this differentiable forward propagation, the gradient can backpropagate to the camera parameters using *PyTorch* [32].

After each iteration of gradient-based optimization, our method recalculates the visibility, as *ShapeAnalyze* and fine-tunes the neural observation field with significant down-sampled points for efficiency, as *LeanNeOF*. The gradient-based optimization stops when the gradient of camera parameters and the difference between two consecutive losses is less than 1e-4. This stopping criterion corresponds to *CheckStepUpdate* in Algorithm. 1.

Non-gradient-based camera placement optimization. The gradient optimization performs well when the object exhibits weak non-convexity. However, objects in real-world scenarios are often diverse and highly non-convex. To escape from local optima, we employ a non-gradient-based optimization method $\mathcal{H}_{non\_grad}(\mathcal{P}, \mathcal{F})$, akin to trust-region optimization. This method fixes the well-optimized camera placements and recalculates the positions and orientations of less optimal cameras.

We update the camera that satisfies two conditions: first, the gradient is less than 1e-4, indicating that the camera has converged; second, the camera still has a large loss after convergence, indicating a better observation camera pose in global space than this one. To replace this camera, we filter the m least-covered regions to calculate the current camera placement neural field. The coverage and observation quality are compared using Equation 8 to generate a new camera pose, set directly above the poorly covered area of the object surface. After generating a new camera pose, we update the neural observation field. After traversing throughout m regions, we choose the best camera pose with the minimum $\mathcal{L}$:

$$p_{w_{new}} = \arg\min_{j} \ \mathcal{L}(\mathcal{P}_{t_{i\to j}}, \mathcal{F}_t)_{j=1:m}. \tag{9}$$

We continue replacing worse camera poses until no better new camera pose can be found.

D. Implementation details

The initialization of camera placements is randomly sampled points within the space of the scene. For differentiable optimization, we use the Adam optimizer to optimize parameters containing both camera poses and the Attention layer. The initial learning rate of the optimizer is $1e^{-3}$, with a gradual decrease during the optimization process. While dealing with occlusion in view, we employ spherical inverse flipping and convex hull construction from [33] to obtain points not hidden by other points in one view, reducing the rendering time and input complexity.

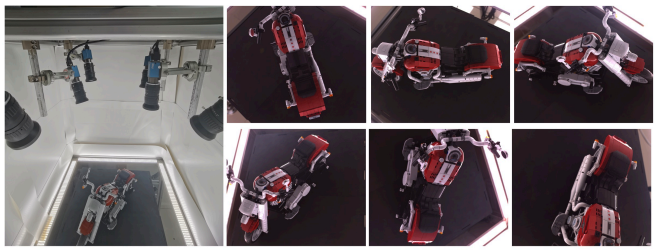


Fig. 4: Camera placement control system. This system is able to control six 4K RGB cameras in SE(3) with a uniform distributed lighting source. Real images captured by optimized cameras.

IV. Experiments

A. Experiment setup

Synthetic environments setup:. The synthetic environment is composed of 2D planar shapes (25), 3D objects (28), and room-scale 3D scenes (8), leading to a total number of 56 which is more than existing methods. The 2D planar shapes are self-generated with random shape (circle, triangular, cube) combinations. For the 3D object models and 3D scenes, we leverage high-quality real scanned reconstruction from Google Scanned Objects (GSO) dataset [12] and Replica dataset [13], respectively.

Real-world experiments setup: We construct a camera placement control system as shown in Fig. 4, composed of six 4K RGB cameras and constant lighting sources. All six cameras can be freely adjusted to capture RGB images. The camera placement can be dynamically adjusted based on the results of camera placement optimization methods. Here, we leverage five diverse objects including the bag, flower, laptop, and motor Lego, for evaluating the performance of methods in the real-world environment.

Baseline methods. Coverage in multi-media placement is an NP-hard problem [18], and various approaches have been proposed to solve this problem. We compare our methods with mainstream methods:

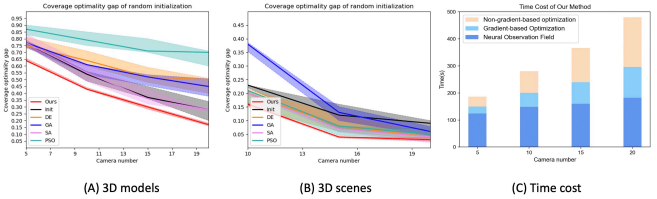
- **Genetic Algorithm (GA)** in [20], emulates natural process to select, generate and evaluate for fitness.
- **Simulated Annealing (SA)** in [10], iteratively replaces inferior solutions until algorithm's temperature reaches a predefined threshold, akin to metal heat treatment.
- **Particle Swarm Optimization (PSO)** in [9], leverages the exchange of information among individuals within a group to facilitate the transition.
- **Differential Evolution (DE)** in [11], distinguishes from GA by crossing with parent individual vectors to generate new ones, proving more effective than GA.
- **Mixed-Integer Programming (MIP)** in [8], employs a branch-and-bound algorithm, solving a sequence of

TABLE I: Comparing Coverage optimality gap and Observation angle quality in 2D Plane and 3D Model datasets. We present the average outcomes for each method across varying numbers of cameras. The best results are highlighted. Here, cam. indicates the number of cameras.

Methods	2D planar shapes								3D objects							
	Coverage optimality gap ↓				Observation angle quality ↑				Coverage optimality gap ↓				Observation angle quality ↑			
	5 cam.	10 cam.	15 cam.	20 cam.	5 cam.	10 cam.	15 cam.	20 cam.	5 cam.	10 cam.	15 cam.	20 cam.	5 cam.	10 cam.	15 cam.	20 cam.
Init	0.77	0.54	0.37	0.28	0	0	0.44	0.69	0.75	0.51	0.34	0.29	0.39	0.93	0.91	0.80
DE[11]	0.75	0.64	0.52	0.50	0.09	0.10	0.58	0.64	0.82	0.67	0.56	0.46	0.34	0.91	0.86	0.88
PSO[34]	0.77	0.61	0.52	0.45	0.02	0.28	0.54	0.52	0.83	0.72	0.31	0.27	0.49	0.88	0.85	0.84
GA [20]	0.87	0.79	0.71	0.70	0.00	0.09	0.37	0.37	0.83	0.76	0.37	0.34	0.41	0.80	0.85	0.86
SA[10]	0.77	0.55	0.36	0.28	0.03	0.15	0.54	0.67	0.76	0.54	0.31	0.24	0.45	0.79	0.81	0.69
MIP[8]	0.68	0.58	0.55	0.53	0.53	0.45	0.40	0.38	0.69	0.52	0.45	0.38	0.62	0.83	0.82	0.81
Ours	**0.64**	**0.43**	**0.30**	**0.17**	**0.57**	**0.54**	**0.73**	**0.72**	**0.63**	**0.40**	**0.28**	**0.20**	**0.70**	**0.96**	**0.91**	**0.89**

TABLE II: Comparing Coverage optimality gap and Observation angle quality in 3D Scene datasetssame as tested in 2D Plane and 3D Model datasets.

Method	Coverage optimality gap ↓			Observation angle quality ↑		
	10 cam.	20 cam.	30 cam.	10 cam.	20 cam.	30 cam.
Init.	0.23	0.12	0.09	0.07	0.07	0.06
DE [11]	0.20	0.08	0.04	0.31	0.26	0.28
GA [20]	0.38	0.13	0.06	0.33	0.37	0.38
PSO [34]	0.21	0.08	0.05	0.55	0.45	0.44
SA [10]	0.20	0.07	0.04	0.44	**0.47**	0.42
Ours	**0.16**	**0.04**	**0.03**	**0.58**	0.46	**0.47**



(A) 3D models (B) 3D scenes (C) Time cost

Fig. 5: The robustness and time cost of our algorithm, we optimize 10 sets of camera placements with different initializations. The solid line is the mean value, with the shading represents the upper and lower bounds. Additionally, we have tested the time cost of different parts in our method.

linear programming derived from original problem.

- **Graph Neural Network (GNN)** in [3] leverages a graph neural network to capture local interactions of the robots on 2D planar shapes.

We compare results with GA, SA, PSO, DE, MIP methods on all 2D planar shapes, 3D objects and room-scale 3D scenes. For GNN, we test only on 2D planar shapes due to the limitation of their method.

Metrics. We plan to evaluate the camera placement in terms of both coverage and observation quality. For coverage evaluation,we consider **Coverage optimality gap** [8], given by the formula:

$$uc = \frac{(K - \sum_{i=0}^{n} cov_i)^2}{K \cdot n^2}, \tag{10}$$

where K represents the required coverage number, (we use 3 in our experiment), n is the number of total visible points, and cov_i represents the number of cameras able to observe point i in current camera placement. Smaller values of the Coverage optimality gap indicate better performance.

For quality evaluation, we consider the Camera-to-object angle and the Camera-to-camera angle as proposed in Section III-B. The angles among vectors from camera rays to the target point within $[45°, 145°]$ has been shown to approximate triangular perception, leading to better observation quality [4]. We calculate the rate of satisfied rays among all camera-to-camera rays as an angle rate to assess the quality, called **Observation angel quality**.

Fig. 6: Comparisons of reconstruction completeness between 3 cameras uniformly distributed and by our optimization method. Missing parts are ones not observed by any camera.

B. Results and analysis

Comparison on 2D planar shapes, 3D objects, Room-scale 3D scenes. We extensively compare our methods in diverse environments (2D planar shapes, 3D objects, and 3D scenes) with different camera numbers. The results of 2D planar shapes and 3D object datasets can be found in Table I, and for 3D scenes, the results are presented in Table II. We use an acceptable number of particles (30) to test all datasets. We also perform extensive experiments with different particle numbers, finding that the methods reach convergence as the particle number increases. While these methods are comparable with ours in terms of performance when converged, their time overhead is at least one hour, which is 48 times longer than ours. The results prove that our method has significant advantages in both coverage and observation quality over existing methods across different camera numbers. Specifically, we find our method showcases stable improvement along with the increase in the number of cameras, constantly showing better performance than existing methods. As for 3D scene datasets, we report the results in Table II, where we continue to achieve state-of-the-art performance compared to all methods. Experiments

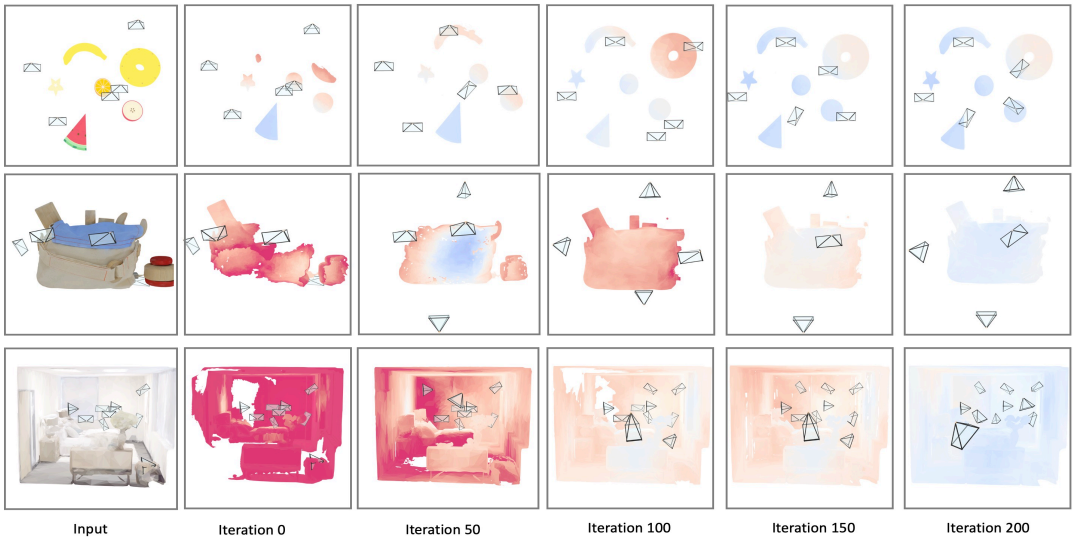


Fig. 7: Visual results of neural observation field during optimization. From left to right, the optimization step increases and the color to blue, the better observation and reconstruction quality.

TABLE III: Comparing single-coverage of learning-based GNN method [3] with our method.We compare results on their self-generated plane dataset with single-coverage metric, the percentage of covered targets number. Here, cam. indicates the number of cameras.

Method	Single-coverage metric [3] ↓			
	10 cam.	20 cam.	30 cam.	40 cam.
GNN [3]	0.41	0.36	0.23	0.14
Ours	**0.36**	**0.33**	**0.21**	**0.13**

on synthetic datasets show the applicability of our approach in various scenarios. The corresponding visual results can be found in Fig. 7, where models from crippled or red to complete and blue, reflecting our optimization process. To exclude the effect of initialization, we perform 10 times experiments with different random initializations to verify the robustness of the approach in Fig. 5. Thin shading bounds of our method illustrate its non-dependency on the quality of initial camera placement. The visualization of reconstruction shown in Fig. 6 as a downstream task, illustrates better completeness and accuracy of reconstructed models achieved by our optimization method.

Comparison with the learning-based method [3]. In addition to traditional methods, we experiment to compare with the learning-based method presented in [3]. Camera poses in [3] can only optimize on 2D planar shapes, thus we adjust ours accordingly. The results are provided in Table III. Our method outperforms the GNN method [3] for various camera placement numbers, which is the state-of-the-art camera placement optimization on 2D planar shapes. However, it's worth noting that our advantage diminishes as the number of cameras increases because it becomes easier to achieve better coverage with more cameras.

Ablation Study. We conduct an ablation study to verify the effectiveness of key components in our method using 2D planar shapes dataset. Optimization targets of all methods are both Coverage optimality gap and Observation angle quality. The results can be found in Table IV, demonstrates better performance than gradient- or non-gradient optimization

TABLE IV: Ablation study of Hybrid optimization and neural observation field. Rows 1-2 use only gradient-based or non-gradient-based optimization methods, and rows 3-5 use discrete or distance-based neural fields as comparison.

Method	Coverage optimality gap ↓	Observation angle quality ↑
Grad. opt. + neural obs. filed	0.510	0.880
Non-grad. opt. + neural obs. field	0.473	0.896
Hybrid opt. + trilinear interpolation	0.588	0.818
Hybrid opt. + neural distance field	0.512	**0.937**
Hybrid opt. + neural obs. field (Ours)	**0.398**	0.933

TABLE V: Real-world experiments on our camera placement control system, we dynamically optimize the placement of six cameras on five real-world models and obtain the coverage and observation quality.

Models	Coverage optimality gap ↓	Observation angle quality↑
Moto	0.06	0.87
Laptop	0.00	0.81
Flower	0.10	0.72
Sandbox	0.00	0.92
Bag	0.03	0.88

(rows 1-2) as the gradient optimization method suffers from a highly non-linearity optimization landscape. While combining it with non-gradient optimization, we observe a significant improvement which proves the effectiveness of using hybrid optimization. Moreover, rows 3-5 demonstrate that our neural observation field outperforms other alternatives, underscoring its capability to provide both differentiable and observation measurements to guide the optimization, resulting in better observation quality that underlies coverage.

Real-world experiments. The real-world experiments conducted using our camera placement control system are summarized in table V. Here, we leverage five real-world models and dynamically adjust the camera placement by our optimization, with input shapes scanned roughly. The results clearly demonstrate that our method still has good performance in real-world environments by almost fully covering the target objects while having good camera quality, capturing information from concave planes of objects as well.

Analysis of Computational Cost. Our method exhibits high efficiency in both learning the neural observation field and the optimization process shown in Fig. 5. Under typical

10-camera seniors, our method requires only 0.08 seconds per iteration for learning the neural observation field and 0.1 seconds for the optimization process in each iteration. Throughout our experiments, our method consistently exhibited the fastest speed (eight times faster than mainstream methods, as reported in [8], [9], [11], [34]), while also achieving state-of-the-art performance.

V. CONCLUSIONS

In this work, we present a novel camera placement hybrid optimization method leveraging the neural observation field. Our method amalgamates the strengths of both gradient-based and non-gradient-based optimization techniques, striking a balance between their respective advantages. To enable a unified observation for both methods, the neural observation field learns the coverage and observation quality of camera placement in a differentiable manner. The results on both synthetic datasets and real-world datasets clearly demonstrate the superiority of our methods. In the future, we would like to exploit our approach under dynamic environments or large-scale scenes *e.g.* a whole building.

REFERENCES

[1] J. Kritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar, "On the real-world applicability of state-of-the-art algorithms for the optimal camera placement problem," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2019, pp. 1103–1108.

[2] Z. Kang and G. Medioni, "Progressive 3d model acquisition with a commodity hand-held camera," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 270–277.

[3] L. Zhou, V. Sharma, Q. Li, A. Prorok, A. Ribeiro, P. Tokekar, and V. Kumar, "Graph neural networks for decentralized multi-robot target tracking," in *Proceedings of the IEEE Conference on Decision Control*, 2022.

[4] P. Rahimian and J. K. Kearney, "Optimal camera placement for motion capture systems," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 3, pp. 1209–1221, 2016.

[5] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine learning for wireless sensor networks security: An overview of challenges and issues," *Sensors*, vol. 22, no. 13, p. 4730, 2022.

[6] T. Zhang, J. Xiao, L. Li, C. Wang, and G. Xie, "Toward coordination control of multiple fish-like robots: Real-time vision-based pose estimation and tracking via deep neural networks." *IEEE CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1964–1976, 2021.

[7] N. allah Mottaki, H. Motameni, and H. Mohamadi, "A genetic algorithm-based approach for solving the target q-coverage problem in over and under provisioned directional sensor networks," *Physical Communication*, vol. 54, p. 101719, 2022.

[8] A. Malhotra, D. Singh, T. Dadlani, and L. Y. Morales, "Optimizing camera placements for overlapped coverage with 3d camera projections," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5002–5009.

[9] V. A. Puligandla and S. Lončarić, "A continuous camera placement optimization model for surround view," *IEEE Transactions on Intelligent Vehicles*, 2023.

[10] B. Shahrokhzadeh, M. Dehghan, and M. Shahrokhzadeh, "Improving energy-efficient target coverage in visual sensor networks," *Journal of Computer & Robotics*, vol. 10, no. 1, pp. 53–65, 2017.

[11] E. O. Rangel, D. G. Costa, and A. Loula, "On redundant coverage maximization in wireless visual sensor networks: Evolutionary algorithms for multi-objective optimization," *Applied Soft Computing*, vol. 82, p. 105578, 2019.

[12] L. Downs, A. Francis, N. Koenig, B. Kinman, R. M. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, 2022.

[13] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[14] L. Gjakova, R. Löser, P. Klimant, and M. Dix, "Decreasing the commissioning time of optical multi-camera inspection systems by simulating surface coverage using the example of formed bipolar plates," *Engineering Proceedings*, vol. 26, no. 1, p. 17, 2022.

[15] L. Zheng, C. Zhu, J. Zhang, H. Zhao, H. Huang, M. Niessner, and K. Xu, "Active scene understanding via online semantic reconstruction," in *Computer Graphics Forum*, vol. 38, no. 7, 2019, pp. 103–114.

[16] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Fusion-aware point convolution for online semantic 3d scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[17] X. Zhang, B. Zhang, X. Chen, and Y. Fang, "Coverage optimization of visual sensor networks for observing 3-d objects: Survey and comparison," *International Journal of Intelligent Robotics and Applications*, vol. 3, pp. 342–361, 2019.

[18] M. Bouhali, A. Bendjeddou, and N. Ghoualmi-Zine, "Classification of coverage algorithms in multimedia wireless sensor networks," in *2021 International Conference on Networking and Advanced Systems (ICNAS)*. IEEE, 2021, pp. 1–5.

[19] T. L. Brown, "Deployment, coverage and network optimization in wireless video sensor networks for 3d indoor monitoring," 2017.

[20] A. H. Navin, B. Asadi, S. H. Pour, and M. K. Mirnia, "Solving coverage problem in wireless camera-based sensor networks by using genetic algorithm," *2010 International Conference on Computational Intelligence and Communication Networks*, pp. 226–229, 2010.

[21] S. W. Feng, K. Gao, J. Gong, and J. Yu, "Sensor placement for globally optimal coverage of 3d-embedded surfaces," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8600–8606.

[22] L. Tao, R. Xia, J. Zhao, F. Wang, and S. Fu, "Research on optimization of multi-camera placement based on environment model," in *Proceedings of the 2023 7th International Conference on Computing and Data Analysis*, 2023, pp. 54–62.

[23] G. Olague, "Automated photogrammetric network design using genetic algorithms," *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 5, 2002.

[24] G. Olague and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern recognition*, vol. 35, no. 4, pp. 927–944, 2002.

[25] M. Saadatseresht, C. S. Fraser, F. Samadzadegan, and A. Azizi, "Visibility analysis in vision metrology network design," *The Photogrammetric Record*, vol. 19, no. 107, pp. 219–236, 2004.

[26] V. A. Puligandla and S. Lončarić, "A multiresolution approach for large real-world camera placement optimization problems," *IEEE Access*, vol. 10, pp. 61 601–61 616, 2022.

[27] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[28] Y. Li, X. Qi, Y. Chen, L. Wang, Z. Li, J. Sun, and J. Jia, "Voxel field fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1120–1129.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] B. Kenwright, "Free-form tetrahedron deformation," in *Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part II 11*. Springer, 2015, pp. 787–796.

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[33] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 24–es.

[34] Z. Jiao, L. Zhang, M. Xu, C. Cai, and J. Xiong, "Coverage control algorithm-based adaptive particle swarm optimization and node sleeping in wireless multimedia sensor networks," *IEEE Access*, vol. 7, pp. 170 096–170 105, 2019.