Boundary Exploration for Next Best View Policy in 3D Robotic Scanning

Leihui Li, Xuping Zhang

Abstract—**The Next Best View (NBV) problem is a pivotal challenge in 3D robotic scanning, with the potential to greatly improve the efficiency of object capture and reconstruction. Current methods for determining the NBV often overlook view overlaps, assume a virtual origin point for the camera's focus, and rely on voxel representations of 3D data. To address these issues and improve the practicality of scanning unknown objects, we propose an NBV policy in which the next view explores the boundary of the scanned point cloud, and the overlap is intrinsically considered. The scanning distance or camera working distance is adjustable and flexible. To this end, a model-based approach is proposed where the next sensor positions are searched iteratively based on a reference model. A score is calculated by considering the overlaps between newly scanned and existing data, as well as the final convergence. Additionally, following the boundary exploration idea, a deep learning network, Boundary Exploration NBV network (BENBV-Net), is proposed, which can be used to predict the NBV directly from the scanned data without requiring the reference model. It predicts the scores for given boundaries, and the boundary with the highest score is selected as the target point of the next best view. BENBV-Net improves the speed of NBV generation while maintaining the performance of the model-based approach. Our proposed methods are evaluated and compared with existing approaches on the ShapeNet, ModelNet, and 3D Repository datasets. Experimental results demonstrate that our approach outperforms others in terms of scanning efficiency and overlap, both of which are crucial for practical 3D scanning applications. The related code is released at github.com/leihui6/BENBV.**

Index Terms—**Next Best View, Point Cloud, Robotic Scanning, Deep Learning**

I. Introduction

The ability to efficiently and automatically scan and reconstruct 3D objects or environments, often referred to as view planning, sensor planning, or active perception, is essential in various applications, including industrial inspection [1], cultural heritage preservation [2], and autonomous robotics [3]. A key challenge is the Next Best View (NBV) problem, which seeks to determine the optimal sequence of views that maximizes the completeness and quality of a 3D scan while minimizing the number of scans required. An efficient NBV policy enables robotic systems, integrated with vision systems, to maximize the coverage of objects or environments with minimal scanning steps, ensuring high-quality 3D reconstructions and benefiting downstream applications. Several constraints are considered in the NBV policy, including object occlusion, unseen region prediction, sensor movement costs, potential offsets, field of view, and resolution of sensors. The NBV problem, proven to be NP-complete when relying on prior geometric knowledge [4], becomes even more challenging when addressing unknown objects or environments, where geometric information is often sparse or incomplete.

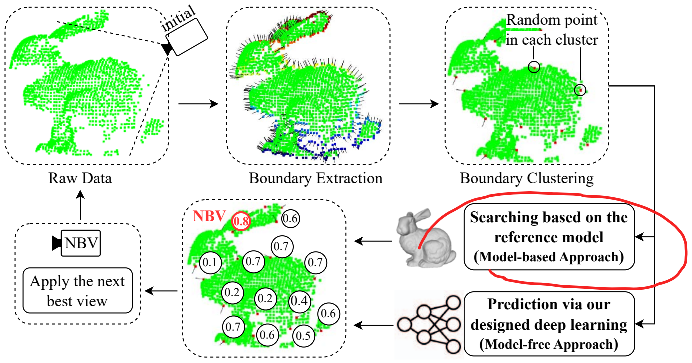

Fig. 1. Overview of our proposed approach: Starting with an initial scan, the boundary of the raw point cloud is detected and clustered to generate potential views. In the model-based approach, the NBV is searched and selected based on the reference model. In the model-free approach, we predict scores for given boundaries and select the one with the highest score to be the NBV.

The NBV problem has been widely explored using 2D images, but 2D approaches are often limited by the lack of depth information and ambiguity in spatial relationships. In contrast, depth data or point clouds inherently capture rich and geometric details about the target and scene, making them more effective for NBV planning in complex environments. NBV methods utilizing 3D data from point clouds can broadly be categorized into model-based and model-free approaches [5]. Model-based methods leverage pre-built or provided reference models of target objects to predict the next best view, with the reference model involved in each step. In contrast, model-free approaches generate the NBV without prior knowledge of the object's reference model, relying solely on the information acquired during the scanning process. In addition, most existing NBV methods utilize mesh or voxel representations of the retrieved 3D point cloud, which are well-suited for large outdoor scenes with low precision requirements. However, these methods require additional preprocessing, such as surface reconstruction and voxelization. These processes increases computational and storage costs while offering lower precision for indoor environments or small objects.

On the other hand, existing methods typically rely on empirically designed view spaces, such as a hemisphere around the target object. However, these approaches limit their generalizability to novel objects. Additionally, they overlook the optimization of the sensor's working distance, assuming

All authors are with the Department of Mechanical and Production Engineering, Aarhus University, Aarhus, Denmark

*Corresponding author: Xuping Zhang, Email: xuzh@mpe.au.dk

the origin point is at the center of the hemisphere, which can lead to variations in imaging quality. Furthermore, while many existing methods assess scanning efficiency based on coverage, they often overlook the significance of overlap between newly acquired data and existing datasets. This overlap is crucial for properly aligning new data with the current dataset, especially in cases where camera movement deviates from the expected position and orientation.

To address these challenges and improve NBV selection performance, we introduce a boundary exploration NBV policy. This approach identifies boundary points from the scanned data as candidate views, where camera positions are defined, and the camera's focus is directed toward the boundaries. First, a model-based method is proposed, which iteratively searches for and selects the NBV with the highest score. This score balances overlap and coverage to optimize scanning efficiency. Second, we present a model-free method utilizing a deep learning framework, the Boundary Exploration NBV Network (BENBV-Net). BENBV-Net predicts scores for the boundary points of the point cloud and selects the point with the highest score as the NBV. This approach eliminates the need for a reference model while improving the inference speed for NBV selection.

In summary, the contributions of this work are summarized as follows:

1) We propose an intuitive NBV policy that explores the boundaries of point clouds directly using the raw data as input. This approach accounts for both the overlap between newly captured and existing data and the overall coverage. Additionally, it allows for an adjustable camera working distance, making it adaptable to various 3D sensors.
2) We propose a model-based approach, where a reference model is used to search and select the view with the highest score as the NBV. Additionally, we introduce a learning-based framework, BENBV-Net, which does not require a reference model. Instead, it takes the scanned point cloud and boundaries as input and predicts scores for each boundary.
3) The efficiency and effectiveness of our approach are thoroughly evaluated against traditional methods on datasets such as ShapeNet, ModelNet, and 3D Repository. Furthermore, the proposed BENBV-Net showcases generalization capabilities on unseen test data and novel objects, achieving performance comparable to the model-based approach while significantly reducing computational time.

II. Related Work

Given 3D data, researchers have explored various approaches to predict the optimal next view for capturing target objects. One such approach [6], [7] involves using mesh representations of the 3D point cloud, where the mesh is utilized by the robot to identify unknown or poorly reconstructed regions. However, this approach incurs additional computational and storage costs beyond the view prediction itself. An alternative strategy uses volumetric representation [8], [9], [10], which divides the scene into multiple voxels, each associated with its observation status. This approach is commonly applied in building octomaps [11], [3] or environmental maps, making it effective for capturing full 3D structures and handling occlusions. However, it comes with trade-offs: volumetric methods are computationally and memory-intensive and may compromise surface detail, especially at lower grid resolutions. To address these limitations, our approach directly processes the raw point cloud, allowing for more accurate and efficient perception of small objects and indoor environments, where high resolution and fine surface details are crucial.

Additionally, the NBV problem in model-free scenarios is particularly challenging due to the lack of prior knowledge about the target object. The difficulty lies in selecting viewpoints that efficiently capture the scene [12], especially when considering time constraints and unexpected occlusions. To address these challenges, various methods have been proposed. For example, the Surface Edge Explorer (SEE) [12], [13], [14] efficiently selects views to improve surface coverage while minimizing movement time. However, it depends on user-defined parameters and struggles with complex geometries. Prediction-guided methods, such as Pred-NBV [15], maximize information gain by intelligently navigating around obstacles, but may face difficulties in dynamic environments. Learning-based methods like PC-NBV [16] and NBV-net [17] introduce deep neural networks that directly process raw point cloud data and current view selection states to predict the information gain of candidate views. However, they are limited by the view space and sensor working distance. Reinforcement learning techniques, such as GenNBV [18] and RL-NBV [19], adaptively explore environments and achieve high coverage ratios, but they require extensive training data, and their selection policies are often not explainable, making the underlying principles unclear.

It is important to note that when minimizing the number of scan views, it is crucial to account for the overlaps between the newly acquired data and existing views [17], [20]. Failing to do so can make it more difficult to accurately register the views [21], potentially leading to errors in the reconstruction process. Additionally, the optimal working distance for the 3D camera must be considered or adjusted flexibly to ensure the best imaging performance and high-quality 3D point clouds.

To this end, we propose an intuitive NBV policy that explores the boundary of the acquired point cloud. The overlap is naturally accounted for, as the next view can always be oriented towards the existing data. Most importantly, our method allows for adjusting the distance between the target surface and the 3D acquisition device to optimize scanning, rather than assuming the object's centroid or original position as the camera's focal point.

III. Problem Definition and Methodology

A. Problem Definition

The Next Best View (NBV) problem in 3D robotic scanning involves determining the next optimal viewpoint to capture data about an object or scene, aiming to enhance the completeness and accuracy of 3D scanning. In our study, the surface data of the object, $s_i \in \mathbb{R}^3$ perceived by the 3D sensor is represented as a point cloud. The view information and the optimization variables are defined as (V_i^{cam}, V_i^{tar}), where $V_i^{cam} \in \mathbb{R}^3$ represents the camera's position, and $V_i^{tar} \in \mathbb{R}^3$ denotes the focal point of the camera. Notably, V_i^{tar} is always located on the boundary points of the point cloud.

Given an initial point cloud s_i, the objective of this paper is to determine (V_i^{cam}, V_i^{tar}) to accelerate the data acquisition process while maximizing the objective function F, as

$$\underset{V_i \in V}{\operatorname{argmax}} F(s_i\left(V_i^{cam}, V_i^{tar}\right)) \tag{1}$$

The F here is defined by the coverage ratio $\mathrm{C} = |s_i|/|S|$ and the overlap ratio $\mathrm{O} = |p_o|/|s_i|$, where S represents the number of points in the reference model, S_i is the number of points in the retrieved point cloud, and $|p_o|$ represents number of overlapping points between the current scanned data and the existing data. In a simulated environment, S is known and predefined, and s_i is generated within a physics-based simulation engine. However, for real-world applications, s_i is derived from actual 3D sensor data, and S is typically unknown in advance due to the novelty of the target object.

B. Framework Overview

Given the captured point cloud, the general idea in this paper is to explore the unseen regions by following its boundary. However, not all boundaries contribute meaningfully to high coverage and overlap. Therefore, we first introduce a model-based approach that requires a reference model, where the NBV is determined through a search process, and the view with the highest score is selected as the next best view. We then present a model-free approach, where we develop and propose a deep learning network that learns the latent space mapping between the designated boundary and the point cloud. This enables the network to predict the most optimal NBV in a time-efficient manner. The overall framework is illustrated in Figure 1.

C. Boundary exploration for NBV policy

The boundaries of retrieved 3D representation data have been examined in previous studies [22], [12], which utilize either triangle surfaces or density-based methods to explore edges. In our study, we compute the boundary directly from the raw point cloud data using the Angle Criterion method proposed by [23] where the angle threshold is set 120 degree. Let B_i represent the boundary points for the i-th scanned data. To reduce the large number of boundary points and simplify the process, we use the K-Means algorithm to cluster the boundary points into 20 clusters. From each cluster, we randomly select one point as V_i^{tar}.

The normals for the scanned data are estimated as $s_i^{N\times 6}$and are used to determine the direction of the camera position. Specifically, the normals are oriented such that their direction aligns toward the camera position, ensuring the angle between the normals and the camera direction is less than 90°. For each V_i^{tar}, we establish a local orthogonal coordinate system (**u**-**v**-**n**). In this system, **n** represents the normal vector at V_i^{tar}, while **u** is defined by

$$\mathbf{u} = V_i^{tar} - \bar{V}_i^{tar} \tag{2}$$

where $\bar{V}_i^{tar}$ represents the centroid of the neighboring points around V_i^{tar}, and **u** indicates the direction of exploration originating from V_i^{tar}. The **v** is then computed by $\mathbf{v} = \mathbf{u} \times \mathbf{n}$. The direction of camera position is expressed as

$$\mathbf{n'} = R_{\mathrm{v}}(\theta) \cdot \mathbf{n} \tag{3}$$

where θ is randomly set to $-45°, 0°$ or $45°$. Given the scanning distance d which is adjustable and can be changed during scanning process, the camera position is determined as

$$V_i^{cam} = d \cdot \mathbf{n'} + V_i^{tar} \tag{4}$$

as illustrated in Figure 2.

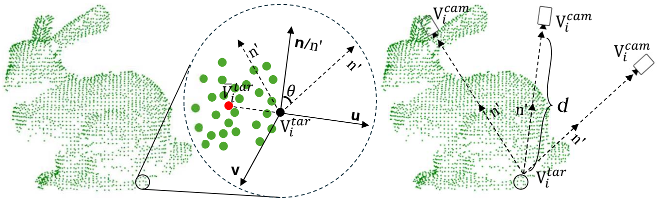

Fig. 2. **u**-**v**-**n**: the V_i^{tar} is marked in black and the $\bar{V}_i^{tar}$ is denoted in red. The camera position is defined on the right, with d indicating the specified distance.

Given the proposed V_i^{cam} and V_i^{tar}, a score is assigned to each potential view by considering both coverage and overlap, and the view with the highest s is selected as the next best view. The score for each view is calculated as

$$s = (1 - W_c) \cdot O_i + W_c \cdot C_i \tag{5}$$

where O_i and C_i represent the overlap and coverage values at the i-th view. The weight W_c is defined as

$$W_c = \frac{1}{1 + e^{-10\cdot(C_i - 0.6)}} \tag{6}$$

The coverage becomes increasingly important, while overlap becomes less significant as the scanning process progresses. Specifically, overlap is weighted more heavily than coverage until the current coverage reaches 60%.

The workflow for searching the NBV is detailed in Algorithm. 1. For the initial view, V_{best}^{cam} is chosen from the surface of a sphere with radius d, while V_{best}^{tar} is set as the origin point $(0,0,0)$. The virtual scanned data can be provided by the simulation engine. The detailed searching process is outlined in Algorithm. 2. The dataset in Algorithm 1 is constructed for the deep learning network we developed, where the proposed views, their corresponding point clouds, and the associated scores are collected.

D. Deep Learning Architecture

The goal of the learning-based approach is to predict the NBV given the detected boundaries, particularly for 3D automatic scanning tasks without requiring a reference model. The training dataset is constructed using the model-based approach and includes the acquired point cloud (**P**), the

Algorithm 1 Workflow for NBV Search

```
P ← ReferencePointCloud
d ← 1.2
$V^{cam}_{best}$, $V^{tar}_{best}$ ← initialView($P$)
TotalScanCount ← 15
i ← 0
existedData ← $None$
Dataset ← [ ]
while $i$ < TotalScanCount do
    scannedData ← simulator($P, V^{cam}_{best}, V^{tar}_{best}$)
    existedData ← existedData + scannedData
    ($V^{cam}_{best}$, $V^{tar}_{best}$), viewList, SList ← searchNBV(...)
    Dataset ← (existedData, viewList, SList)
end while
```

Algorithm 2 SearchNBV

```
Input : P, existedData, d
B ← BoundaryPointsDetection(existedData)
$\{B_i\}$ ← BoundaryPointsCluster(B, K = 20)
$\{(V_i^{cam}, V_i^{tar})\}$ ← GenerateViews($B_i$, d)
viewList ← $\{(V_i^{cam}, V_i^{tar})\}$
MaxS ← 0; MaxIndex ← 0; SList ← []
for ($V_i^{cam}, V_i^{tar}$) ∈ viewList do
    C, O = simulator(P, existedData, $V_i^{cam}, V_i^{tar}$)
    S ← $(1 - W_c) \cdot O + W_c \cdot C$
    if S > MaxS then
        MaxS ← S; SList.add(S)
        MaxIndex ← i
    end if
end for
Return viewList[MaxIndex], viewList, SList
```

detected boundaries (**B**), and their corresponding scores. To enable the network to efficiently learn to select the best view, the training supervision pair for the BENBV-Net is defined as (P, B, C) where **C** consists of the density at **B** and view order.

We propose BENBV-Net, a deep neural network for hierarchical feature learning and NBV score prediction from 3D point clouds. The network employs two parallel encoders: a point feature encoder for spatial coordinates and a normal feature encoder for surface normals, both utilizing convolutional layers with batch normalization and ReLU activation. The **P** feature follows a structure similar to PointNet, processing point cloud data in a permutation-invariant manner. Boundary features are extracted via fully connected layers, while contextual features, including point density and view order, are fused using a learnable density and view order fusion module. These features are combined with global representations through residual blocks for multi-scale refinement. A multi-head self-attention mechanism captures long-range dependencies.

The prediction head computes per-point NBV scores using fully connected layers with dropout regularization. The entire network is trained end-to-end from raw point cloud data. An overview is shown in Figure 3.

A position-aware loss function for next-best-view prediction is designed which emphasizes predictions at the score of different boundaries. The loss function L for a batch of predictions y_s^i and ground truth values Y_s^i is defined as:

$$L(y_s, Y_s) = \lambda \sum (w_i (y_s^i - Y_s^i)^2) \tag{7}$$

where $w_i = ((x_i - 12)/10)^2 + 0.3$ is the position-dependent weight for the i-th view position, x_i is the view order index within the range $[0, 19]$, and λ is a scaling factor set to 5.0. This weighting scheme emphasizes predictions at extreme viewing scans, as positions at the beginning are typically more critical and informative for NBV selection than central views. Specifically, the minimum weight is 0.3 for positions near the middle, with front scans receiving higher importance than those in the latter half of the sequence.

IV. Implementation Details and Experiments

A. Experiment Setup

We use PyBullet [24] as the simulation platform, with the camera configuration set to a near distance of 0.01 m, a far distance of 5 m, and a captured image size of 1280 × 720 pixels, with a 70° field of view (FOV). The scanning distance for the 3D camera is set to 1.2 meters, which is an optimal distance for most low-cost cameras. The training of BENBV-Net and all experiments were conducted on a system running Ubuntu 24.04, equipped with an Intel i9 processor and an NVIDIA RTX 3090 GPU.

B. Dataset

We evaluate and train our model-based method and BENBV-Net using the ShapeNet [25], ModelNet40 [26] datasets, and 3D Repository from Stanford University [1] and Georgia Tech [2], with an example shown in Figure 4. The ShapeNet and ModelNet40 datasets are used for both training and evaluation, while the 3D Repository is reserved exclusively for testing. Specifically, we use ShapeNetV1, selecting 20 models per category for training and 15 models per category for testing. Similarly, for ModelNet40, which includes 40 categories of CAD-generated meshes, we select 20 models per category for training and 15 for testing. For each model, 16 scans are extracted as trainable data. This setup results in a total of over 24,000 training samples from ShapeNetV1 and ModelNet40, 18,000 testing samples, and 128 testing samples from the 3D Repository. The datasets we built and utilized are available online[3]. The model used in the simulation environment for scanning is sampled as points using Poisson Disk Sampling [27].

C. Network Training

Our proposed BENBV-Net is trained using the Adam optimizer with a base learning rate of 0.001 and a mini-batch size of 128. During both training and testing, the input data are randomly downsampled to 4,096 points. The loss converges to 0.002 after approximately 150 epochs. To enhance model

[1] graphics.stanford.edu/data/3Dscanrep/

[2] sites.cc.gatech.edu/projects/large_models

[3] huggingface.co/datasets/Leihui/NBV

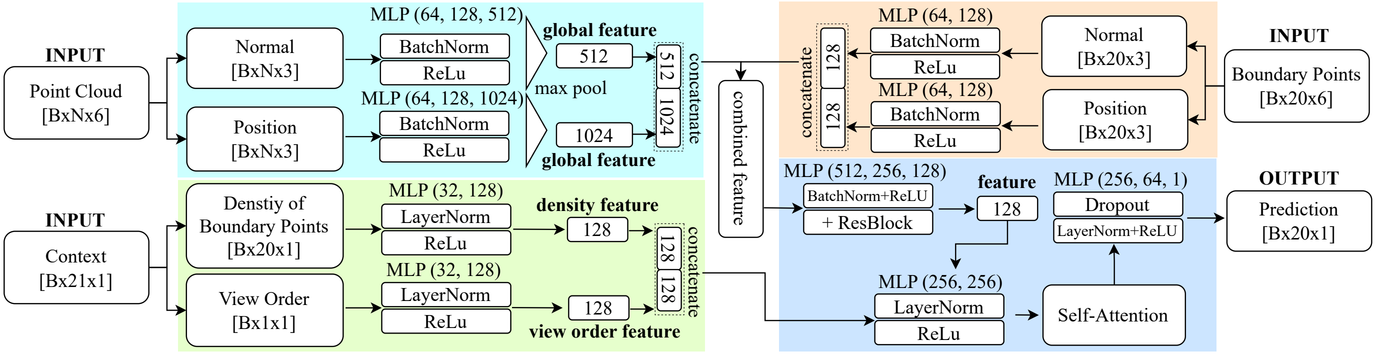

Fig. 3. Our developed deep learning framework, BENBV-Net, for the NBV policy takes the raw point cloud, boundary points, and the corresponding context as input. The network first extracts the global feature of the current point cloud, which is then fused with the global feature from the boundary points, along with the features extracted from the context. These combined features are used to predict a score for each boundary point, with the highest-scoring boundary selected as the NBV.

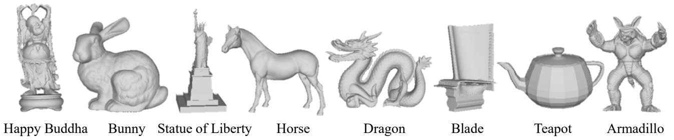

Fig. 4. An example from the 3D Repository dataset, which consists of classic 3D data widely used in the 3D research field.

generalization, the training dataset is augmented by applying random rotations with angles ranging from -10° to 10° along the X, Y, and Z axes. The entire training process takes approximately 2 hours.

D. Evaluation Metrics

The performance of the comparison methods is evaluated using the following metrics: 1) final coverage, 2) overlap between the current view and existing data, 3) Chamfer Distance (CD), 4) Hausdorff Distance (HD), and 5) scanning efficiency. Specifically, we calculate the average of the initial overlap values, as early overlaps are more critical and challenging to optimize in practical scenarios. Furthermore, we evaluate scanning quality by comparing the final retrieved points with the original model using Chamfer and Hausdorff distances. The Chamfer Distance measures the average distance between two point sets, providing an overall indication of their similarity, while the Hausdorff Distance captures the maximum distance from any point in one set to its nearest neighbor in the other, emphasizing worst-case alignment errors. Additionally, scanning efficiency, which aligns with the objectives of the NBV policy, is evaluated by

$$e = \mathrm{c} * 100/\mathrm{v} \tag{8}$$

where v represent the number of views required to achieve 90% coverage, and $\mathrm{c} \in [0,1]$ denote the final coverage ratio. A higher e value indicates greater efficiency in the NBV policy. Specifically, it reflects the ability of the policy to achieve a high coverage with fewer views.

E. Performance Comparison

We compare our proposed method with approaches that utilize point clouds as input data, including PC-NBV [16], a deep network for next-best-view planning, and SEE [12], a density-based edge exploration method for NBV planning. For the comparison, we use the pretrained model provided by PC-NBV and the default parameter settings for small objects as outlined in SEE's work. Additionally, the following policies are used as baselines:

1) **Random Boundary**: The next best view is randomly chosen from the detected boundary of the retrieved point cloud.
2) **Random Sphere**: The next camera position is randomly generated on the surface of a sphere.
3) **Random Uniform Sphere (Random U- Sphere)**: The next camera position is randomly selected from uniformly distributed points on a sphere.
4) **Ours (BENBV)**: It searches and selects the optimal next view given the detected boundary points..
5) **Ours via Deep Learning (BENBV-Net)**: Using the detected boundary and the captured point cloud, it predicts the score of the next best view directly without a search process.

As shown in Tables I, II, and III, we evaluate the performance of our NBV policy in comparison to other methods on the ShapeNet, ModelNet, and 3D Repository datasets. Specifically, each method is run 60 times on the 3D Repository dataset and twice on ShapeNet and ModelNet. The initial view is provided randomly on a spherical surface. The reported coverage represents the final coverage achieved after 15 scanning attempts, while the overlap is achieved during the first five scans. The best results are marked in bold and second best result are underlined.

The results show that our method achieves final coverage rates of 89% and 95% on the test datasets, surpassing all compared approaches. The scanning efficiency is approximately 9.0 and 13.0, outperforming other traditional methods. Additionally, our approach maintains an average overlap of 50% to 60%, significantly higher than the 30% achieved by PC-NBV. The overlap and efficiency of the random boundary method are higher than those of other random-based methods, indicating that exploring based on the boundary better considers overlap and final coverage compared to random methods that limit the view space. Finally, the Chamfer and Hausdorff distances used to evaluate the quality of the reconstructed 3D scans are either lower than or comparable to those of other methods, further validating the accuracy of our approach.

TABLE I
EVALUATION RESULTS OF NEXT-BEST-VIEW POLICIES ON SHAPENET, THE NUMBER OF VIEWS IS SET TO 15 MAXIMUM. THE SYMBOLS ↑ AND ↓ INDICATE WHETHER A HIGHER OR LOWER VALUE IS BETTER, RESPECTIVELY.

NBV Policy	ShapeNet				
	Coverage (%) ↑	Overlap (%) ↑	CD (mm) ↓	HD (mm) ↓	Efficiency ↑
BENBV	**89.09**	**59.29**	0.38	**9.25**	**8.76**
Random Boundary	86.27	54.38	0.39	10.44	6.58
Random Sphere	86.01	36.03	0.36	10.06	6.34
Random U- Sphere	86.41	34.62	0.36	10.05	6.48
PC-NBV	87.36	33.81	**0.34**	9.78	7.18
SEE	62.89	55.21	1.26	17.62	4.13
BENBV-Net	85.92	55.25	0.51	10.91	7.51

TABLE II
EVALUATION RESULTS OF NBV POLICIES ON MODELNET40.

NBV Policy	ModelNet				
	Coverage (%) ↑	Overlap (%) ↑	CD (mm) ↓	HD (mm) ↓	Efficiency ↑
BENBV	**89.07**	**62.21**	0.39	8.81	**9.01**
Random Boundary	87.77	53.80	0.32	8.86	7.12
Random Sphere	87.87	37.04	0.28	8.30	6.97
Random U- Sphere	87.31	34.30	0.29	8.53	6.81
PC-NBV	88.19	33.05	**0.27**	**8.21**	7.49
SEE	65.80	56.22	1.24	17.29	4.40
BENBV-Net	87.34	58.17	0.41	9.43	8.03

TABLE III
EVALUATION RESULTS OF NBV POLICIES ON 3D REPOSITORY.

NBV Policy	3D Repository				
	Coverage (%) ↑	Overlap (%) ↑	CD (mm) ↓	HD (mm) ↓	Efficiency ↑
BENBV	**95.04**	53.52	**0.11**	**5.18**	**13.00**
Random Boundary	93.82	**57.14**	0.12	6.13	9.03
Random Sphere	87.88	32.31	0.21	7.11	6.52
Random U- Sphere	85.49	24.97	0.25	7.45	5.82
PC-NBV	91.93	32.57	0.14	6.22	8.59
SEE	77.73	57.87	0.50	12.54	5.53
BENBV-Net	94.25	47.01	0.12	5.90	11.23

Using 50%, 80%, and 90% coverage as milestones for evaluating the NBV policy, the number of views required to reach each milestone is detailed in Table IV. The results indicate that random-based methods, such as Random Uniform Sphere and PC-NBV, achieve 50% coverage within four scans but require significantly more scans, such as at least 12 for the ShapeNet dataset and 10 for the 3D Repository, to reach 90% coverage. In contrast, BENBV method achieves these milestones more efficiently. For instance, it requires approximately 6 views to reach 80% coverage and 7 views to achieve 90% coverage on dataset 3D repository. Additionally, BENBV-Net demonstrates comparable performance, achieving higher convergence than PC-NBV with fewer scans. Therefore, the results validate the effectiveness of our method, including the model-based approach and BENBV-Net, in rapidly achieving high coverage with fewer scanning views.

TABLE IV
COMPARISON OF NBV POLICIES BASED ON THE NUMBER OF VIEWS REQUIRED TO ACHIEVE 50%, 80%, AND 90% COVERAGE.

NBV Policy	ShapeNet			ModelNet40			3D Repository		
	Coverage								
	50%	80%	90%	50%	80%	90%	50%	80%	90%
BENBV	4.70	**8.21**	**10.17**	4.41	**8.10**	**9.89**	3.83	**6.40**	**7.31**
Random Boundary	4.64	10.66	13.11	4.27	9.89	12.32	3.76	8.31	10.39
Random Sphere	4.59	10.90	13.57	4.33	10.01	12.60	4.69	10.59	13.48
Random U- Sphere	4.52	10.60	13.34	4.40	10.10	12.81	5.34	11.85	14.69
PC-NBV	**3.55**	9.12	12.17	**3.53**	8.90	11.78	3.23	7.61	10.70
SEE	7.23	14.03	15.22	6.34	13.12	14.94	4.03	10.46	14.05
BENBV-Net	4.38	8.91	11.44	4.12	8.52	10.88	**3.20**	6.50	8.39

Furthermore, the detailed overlap metrics are outlined in Table V, where we present the overlap during the first 3 scans, scans 3 to 6, and scans 6 to the final attempt as intervals. According to the results, BENBV generally achieves the highest overlap performance, such as 54% on ShapeNet and 57% on ModelNet for the first 3 scans. In addition, BENBV-Net achieves the second-best overlap, with 49% and 52% on the same datasets. The PC-NBV method has the lowest overlap during the first 3 scans, with averages of 24%, 26%, and 23% on the tested datasets. However, BENBV-Net can achieve an average overlap of 46%, 65%, and 80% on the given datasets. It demonstrates that the effectiveness of PC-NBV lies in its approach of ignoring overlap and focusing solely on the high unseen regions during the view selection process. The SEE method achieves the highest overlap between scans 3 and 6 but struggles to identify views that significantly enhance coverage. This is evident in Table IV, where SEE exhibits the lowest scanning efficiency.

TABLE V
COMPARISON OF NBV POLICIES BASED ON OVERLAP RETRIEVED DURING THE FIRST 3 SCANS, SCANS 3 TO 6, AND SCANS 6 TO THE FINAL VIEW.

NBV Policy	ShapeNet			ModelNet40			3D Repository		
	Scan View								
	1 - 3	3 - 6	6 - 15	1 - 3	3 - 6	6 - 15	1 - 3	3 - 6	6 - 15
BENBV	**54.95**	68.38	**85.48**	**57.40**	71.59	**85.64**	48.45	65.85	**86.11**
Random Boundary	49.21	63.29	72.37	48.79	63.33	71.64	**51.50**	67.06	77.75
Random Sphere	28.91	49.33	61.37	30.80	47.68	59.50	25.01	45.43	59.06
Random U- Sphere	27.50	47.36	58.53	27.82	45.37	57.56	18.80	36.22	51.43
PC-NBV	24.49	49.04	65.41	25.90	45.74	63.81	22.68	50.50	69.22
SEE	44.69	**72.33**	76.20	45.28	**73.94**	79.13	44.90	**78.36**	82.55
BENBV-Net	49.62	66.48	81.53	52.47	68.67	81.54	39.09	62.73	79.38

To provide a clearer understanding of the compared NBV policies, the convergence of each attempt and the overlap between views are illustrated using the 3D repository dataset as an example, as shown in Figure 5 and Figure 6. These figures demonstrate that our method effectively considers both overlap and coverage. First, BENBV achieves the highest coverage compared to other methods, except for the *Blade* data, which has an internal composition that limits observability and prevents further coverage improvement. Second, both the model-based method and BENBV-Net consider overlap. BENBV-Net performs better in terms of overlap for the *Statue of Liberty* dataset, demonstrating its ability to generalize to novel objects, which is a challenge even for the model-based method.

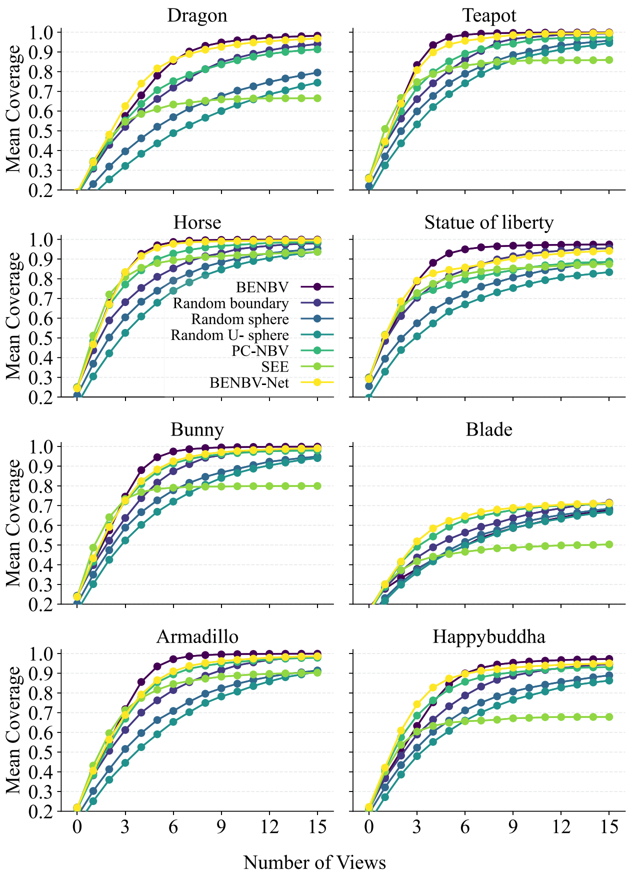

Fig. 5. Coverage for each view in the 3D Repository, with the first view (0-th view) initialized by random scanning based on a sphere surface.

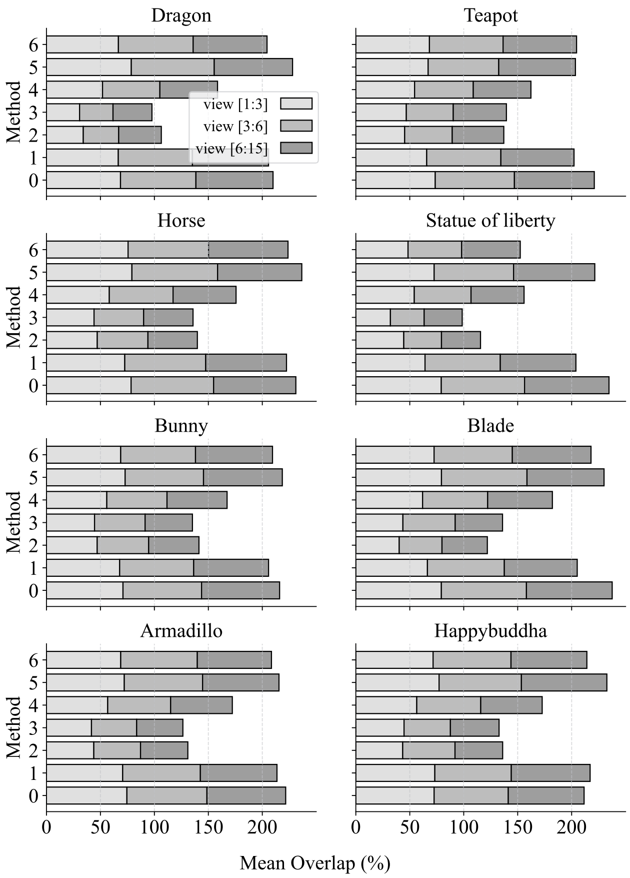

Fig. 6. Overlap during the first 3 scans, scans 3 to 6, and scans 6 to final in the 3D Repository dataset, where the methods are labeled as follows: 0: BENBV-Net; 1: SEE; 2: PC-NBV; 3: Random uniform sphere; 4: Random sphere; 5: Random boundary; 6: BENBV.

A significant drawback of the model-based method is its increased processing time, as illustrated in Table VI, which presents the average time spent scanning each object in the tested datasets. The results reveal that the model-based approach (BENBV) requires approximately 8 to 13 times more time than other methods. In contrast, our learning-based approach, BENBV-Net, takes only 7.8 seconds, demonstrating comparable efficiency in terms of time consumption.

TABLE VI
THE AVERAGE PROCESS TIME (SECONDS) FOR EACH OBJECT IN TESTED DATASETS.

	BENBV	Random Boundary	Random Sphere	Random U- Sphere	PC-NBV	SEE	BENBV-Net
ShapeNetV1	63.2	7.2	4.6	4.6	5.3	6.9	7.8
ModelNet40	66.6	8.1	5.8	5.6	6.4	7.9	7.6
Stanford 3D	64.0	7.0	6.9	6.2	5.9	5.3	7.8

In summary, compared to the tested random-based approaches, SEE, and PC-NBV, the random-based methods can achieve moderate final coverage but require more views to reach 90% coverage. PC-NBV achieves high coverage rapidly but neglects coverage during data acquisition and shows the lowest overlap among the tested methods. SEE provides high overlap during scanning but struggles to identify views that significantly increase coverage. The experiments demonstrate that our method achieves higher completeness more quickly while also considering the overlap between views.

A more specific demonstration is illustrated in Figure. 7, where the object shown is the *bunny*. The same initial view is provided, with the initial coverage at 28%. The coverage (C) and overlap (O) are shown at the 3rd, 6th, and 10th scans. It demonstrates that BENBV-Net achieves the highest overlap, while maintaining a coverage higher than the other methods.

V. Conclusion and Future Work

In this paper, we propose an NBV policy that explores the boundary of the point cloud while balancing coverage and overlap. We present two approaches: a model-based approach, BENBV, which uses a reference model to calculate scores for potential views based on overlap and coverage, selecting the view with the highest score as the NBV. The second approach, BENBV-Net, is learning-based and uses a deep learning network trained on previously collected data. It predicts scores directly from the scanned data and proposed views, allowing NBV selection without the need for a reference model.

The experiments demonstrate that both of our proposed methods achieve high coverage and overlap when evaluated on public datasets such as ShapeNet, ModelNet40, and the 3D Repository. Furthermore, the time efficiency of our learning-based method is comparable to traditional approaches, underscoring its practicality for 3D scanning tasks.

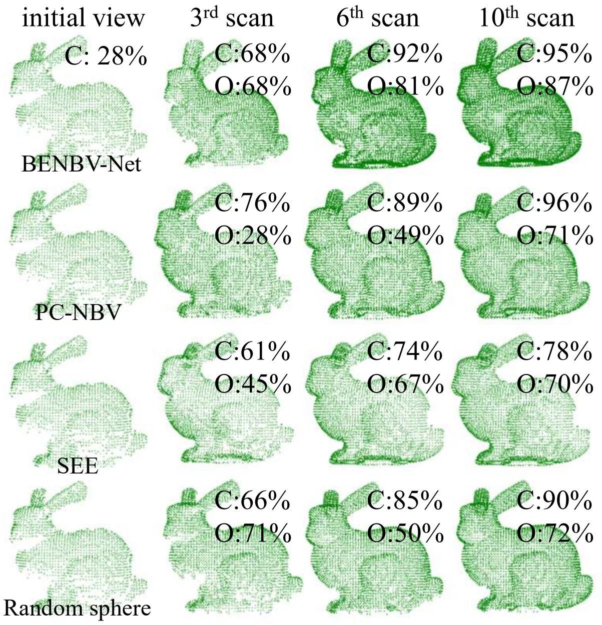

Fig. 7. An example of the NBV policies is illustrated using a bunny object.

However, the method is not fully end-to-end, as it relies on boundary extraction to predict scores. Additionally, future work will address this limitation by exploring the incorporation of action space for robotic movement.

Acknowledgments

The support of Jungner Company is gratefully acknowledged.

References

[1] Jose Luis Alarcon-Herrera, Xiang Chen, and Xuebo Zhang. Viewpoint selection for vision systems in industrial inspection. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4934–4939. IEEE, 2014.

[2] Nikolaos Giakoumidis and Christos-Nikolaos Anagnostopoulos. Arm4ch: A methodology for autonomous reality modelling for cultural heritage. *Sensors*, 24(15):4950, 2024.

[3] Ana Batinovic, Antun Ivanovic, Tamara Petrovic, and Stjepan Bogdan. A shadowcasting-based next-best-view planner for autonomous 3d exploration. *IEEE Robotics and Automation Letters*, 7(2):2969–2976, 2022.

[4] Glenn H Tarbox and Susan N Gottschlich. Planning for complete sensor coverage in inspection. *Computer vision and image understanding*, 61(1):84–111, 1995.

[5] Inhwan Dennis Lee, Ji Hyun Seo, and Byounghyun Yoo. Autonomous view planning methods for 3d scanning. *Automation in Construction*, 160:105291, 2024.

[6] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10:611–631, 2015.

[7] Luca Morreale, Andrea Romanoni, and Matteo Matteucci. Predicting the next best view for 3d mesh refinement. In *Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15*, pages 760–772. Springer, 2019.

[8] Stéphanie Aravecchia, Antoine Richard, Marianne Clausel, and Cédric Pradalier. Next-best-view selection from observation viewpoint statistics. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10505–10510. IEEE, 2023.

[9] Xiaotong Yu and Chang Wen Chen. Semantic-aware next-best-view for multi-dofs mobile system in search-and-acquisition based visual perception. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3713–3721, 2024.

[10] Menaka Naazare, Francisco Garcia Rosas, and Dirk Schulz. Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot. *IEEE Robotics and Automation Letters*, 7(2):3779–3786, 2022.

[11] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34:189–206, 2013.

[12] Rowan Border and Jonathan D Gammell. The surface edge explorer (see): A measurement-direct approach to next best view planning. *The International Journal of Robotics Research*, page 02783649241230098, 2024.

[13] Rowan Border, Jonathan D Gammell, and Paul Newman. Surface edge explorer (see): Planning next best views directly from 3d observations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6116–6123. IEEE, 2018.

[14] Rowan Border and Jonathan D Gammell. Proactive estimation of occlusions and scene coverage for planning next best views in an unstructured representation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4219–4226. IEEE, 2020.

[15] Harnaik Dhami, Vishnu D Sharma, and Pratap Tokekar. Pred-nbv: Prediction-guided next-best-view planning for 3d object reconstruction. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7149–7154. IEEE, 2023.

[16] Rui Zeng, Wang Zhao, and Yong-Jin Liu. Pc-nbv: A point cloud based deep network for efficient next best view planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7050–7057. IEEE, 2020.

[17] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.

[18] Xiao Chen, Quanyi Li, Tai Wang, Tianfan Xue, and Jiangmiao Pang. Gennbv: Generalizable next-best-view policy for active 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16436–16445, 2024.

[19] Tao Wang, Weibin Xi, Yong Cheng, Hao Han, and Yang Yang. Rl-nbv: A deep reinforcement learning based next-best-view method for unknown object reconstruction. *Pattern Recognition Letters*, 2024.

[20] Richard Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on pattern analysis and machine intelligence*, 21(10):1016–1030, 1999.

[21] Sara Hatami Gazani, Matthew Tucsok, Iraj Mantegh, and Homayoun Najjaran. Bag of views: An appearance-based approach to next-best-view planning for 3d reconstruction. *IEEE Robotics and Automation Letters*, 9(1):295–302, 2023.

[22] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *2011 IEEE International Conference on Robotics and Automation*, pages 4869–4874. IEEE, 2011.

[23] Gerhard H Bendels, Ruwen Schnabel, and Reinhard Klein. Detecting holes in point set surfaces. 2006.

[24] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[25] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[26] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[27] Cem Yuksel. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum*, volume 34, pages 25–32. Wiley Online Library, 2015.