MAP-NBV: Multi-agent Prediction-guided Next-Best-View Planning for Active 3D Object Reconstruction

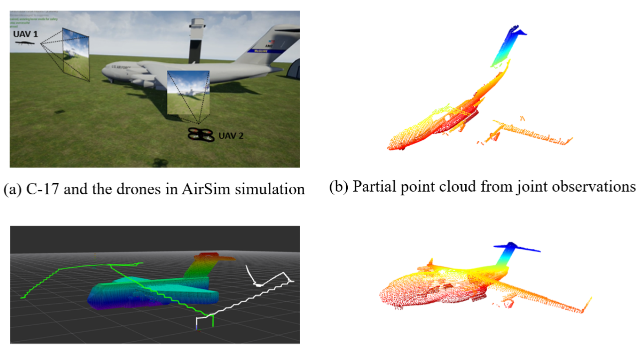
Harnaik Dhami*   Vishnu D. Sharma*   Pratap Tokekar

***Abstract*— We propose *MAP-NBV*, a prediction-guided active algorithm for 3D reconstruction with multi-agent systems. Prediction-based approaches have shown great improvement in active perception tasks by learning the cues about structures in the environment from data. But these methods primarily focus on single-agent systems. We design a next-best-view approach that utilizes geometric measures over the predictions and jointly optimizes the information gain and control effort for efficient collaborative 3D reconstruction of the object. Our method achieves 22.75% improvement over the prediction-based single-agent approach and 15.63% improvement over the non-predictive multi-agent approach. We make our code publicly available through our project website: http://raaslab.org/projects/MAPNBV/**

I. Introduction

Visual surveying and inspection with robots have been studied for a long time for a wide range of applications such as inspection of civil infrastructure [1], [2] and large vehicles [3], [4], precision agriculture [5], and digital mapping for real estate [6], [7]. The utilization of robots in these applications is highly advantageous as they can access hard-to-reach areas with greater ease and safety compared to situations with direct human involvement. Recent work on making robots autonomous for these tasks make their use more appealing. This work focuses on one such long-studied problem of 3D object reconstruction [8], where the objective is to digitally reconstruct the object of interest by combining observations from multiple vantage points. While it could be easier to achieve this in an indoor environment by carefully placing sensors around the object, the same can't be achieved for the outdoors and open areas. For the latter, the sensor(s), must be moved around the object to capture information from different viewpoints. This can be realized with sensors such as cameras and LiDARs mounted on unmanned aerial vehicles (UAVs). A UAV with unlimited power supply capacity could capture infinite observations for an almost perfect reconstruction of the object, but the real-world limitation of battery capacity adds another dimension to the problem: achieving an accurate 3D reconstruction as fast as possible.

The trade-off between reconstruction accuracy and task duration in unknown environments is commonly addressed through Next-Best-View (NBV) planning, wherein a robot determines the optimal location for the next observation to maximize information gain. Numerous solutions have been proposed by the research community to tackle this problem, with a majority of them catering to single-agent systems [9]. However, deploying a team of robots instead of a single agent can enhance task efficiency multi-fold, while also offering additional benefits such as fault tolerance through redundancy. But the direct application of single-agent NBV methods to multi-agent systems does not translate well in terms of performance. This issue stems from the potential overlap between the individual observations. An efficient multi-agent NBV formulation requires coordination among robots to build a joint representation and minimize the overlap.



(a) C-17 and the drones in AirSim simulation (b) Partial point cloud from joint observations

(c) Flight path prescribed by MAP-NBV (d) Final point cloud after MAP-NBV termination

Fig. 1: Application of *MAP-NBV* to a C-17 plane.

In this work, we extend our previous work on prediction-driven single-agent NBV, Pred-NBV [10], to a team of robots for 3D reconstruction to bring the advantages of the prediction-guided approach to a multi-agent system. We call this multi-agent prediction-based next-best-view method MAP-NBV. Pred-NBV [10] uses a 3D point cloud prediction network along with a geometric NBV approach while also considering the control effort required for object reconstruction. An important feature of Pred-NBV is that it doesn't require the partially observed point cloud to be centered at the full object center, an implicit assumption in many 3D reconstruction networks. Naively extending Pred-NBV to a team of robots would result in significant overlap as all the agents would move in the same direction to maximize individual information gain. This is inefficient as it would be more advantageous for the robots to move in different directions. MAP-NBV solves this issue by defining NBV measures over joint observation. We accomplish this by removing duplicate points in observations from multiple robots when calculating the information gain. Along with

*Equal contribution. Names are listed alphabetically.

Authors are with the Department of Computer Science, University of Maryland, U.S.A. {dhami, vishnuds, tokekar}@umd.edu.

This work is supported by the ONR under grant number N00014-18-1-2829.

this, we account for the total control effort in our NBV objective, which results in efficient planning for the whole team.

We make the following contributions in this work:

1) We propose a multi-agent, prediction-based NBV planning approach for active 3D reconstruction of various objects with a novel objective combining visual information gain and control effort.
2) We modify a single-agent baseline NBV algorithm based on [11] that uses frontier-based information gain, and extend its functionality to effectively operate in multi-agent settings.
3) We show that our method outperforms Pred-NBV [10], a single-agent prediction-based algorithm, by **22.75%** and the multi-agent version of a traditional NBV baseline [11] by **15.63%**.

We share the qualitative results and release the project code from our method on our project website[1].

II. Related Work

The use of robots for data acquisition purposes is an extensively studied topic for various domains. Their usage range from infrastructure inspection [12] and environment monitoring [13], [14] for real-world application to the real-world digitization for research datasets and simulations [6], [7], [15]. When the environment is unknown, active methods such as next-best-view (NBV) are used to construct an object model on the fly by capturing additional observations. A majority of the works on NBV planning use information-theoretic measures [9] for selection to account for uncertainty in observations [9], [16], [17]. The widely used frontier and tree-based exploration approaches also utilize uncertainty about the environment for guiding the robot motion [18]–[21]. Some works devise geometric methods which make inferences about the exact shape of the object of interest and try to align the observations with the inferred model [22]–[24]. Prediction-based NBV approaches have emerged as another alternative in recent years, where a neural network takes the robot and/or the environment state as the input and NBV pose or velocity as the output [10], [25]–[27].

A majority of the existing work on NBV is focused on single robot systems. The task performance can be enhanced by adding more robots to the systems, but directly extending single-robot NBV approaches to multi-robot systems may result in sub-optimal performance due to significant overlap in observations. This issue led to the development of exploration algorithms specifically for multi-robot systems [28]–[30] with information-theoretic measures for determining NBV. Some recent works on multi-robot systems have explored the utilization of predictions for improvement in task efficiency. Almadhoun et al. [31] designed a hybrid planner that switches between a classical NBV approach and a learning-based predictor for NBV selection but uses a partial model obtained by robot observations only. Wu et al. [32] use a point cloud prediction model for plants to use the predicted point cloud as an oracle leading to better results than the traditional approaches. This method uses entropy-based information gain measures for NBV and is designed for plant phenotyping with robotic arms. These methods do not consider the control effort required which is important for UAVs with energy constraints when deployed for observing large objects such as airplanes and ships. Also, these works employ information theoretic NBV approaches. We aim to explore a prediction-based approach for geometric NBV selection.

In this work, we extend Pred-NBV [10] which also uses point cloud prediction and build a multi-robot NBV planner. The prediction on the point cloud makes the pipeline modular and interpretable and can be improved by improving individual modules. We select NBV based on information gain, as well as control effort, making our approach more grounded in real-world limitations.

III. Problem Formulation

We are given a team of n robots, each equipped with a 3D sensor. The team flies around a closed object of volume $\mathcal{V} \in \mathbb{R}^3$ and observes the point on its surface $\mathcal{S} \subset \mathcal{V}$. The surface points s_i observed by the robot r_j from the view-point $\phi_k \in \Phi$ are represented as a voxel-filtered point cloud and the relationship between them is defined as $s_i = f(r_j, \phi_k)$. The robot r_j follows a trajectory ξ_{r_j}, consisting of multiple viewpoints, and keeps track of the points observed so far. The distance traveled by a robot between two poses ϕ_i and ϕ_j is represented by $d(\phi_i, \phi_j)$. The point cloud observed by the team of robots is the union of the surface points observed by the individual robots over their respective trajectories, i.e., $s_{\bar{\xi}} = \bigcup_{i=1}^{n} \bigcup_{\phi \in \xi_{r_i}} f(r_i, \phi)$ and $\bar{\xi}$ represents the set of trajectories for each robot, i.e., $\bar{\xi} = \{\xi_{r_1}, \xi_{r_2}, ..., \xi_{r_n}\}$.

The objective is to find a set of feasible trajectories $\bar{\xi}^* = \{\xi^*_{r_1}, \xi^*_{r_2}, ..., \xi^*_{r_n}\}$, such that the team observes the whole voxel-filtered surface, while also minimizing the total distance traveled by the robots on their respective trajectories.

$$\bar{\xi}^* = \arg\min_{\bar{\xi}} \sum_{i=1}^{n} \sum_{j=1}^{|\xi_{r_j}|-1} d(\phi_j, \phi_{j+1}) \tag{1}$$

$$\textit{such that} \quad \bigcup_{i=1}^{n} \bigcup_{\phi \in \xi_{r_i}} f(r_i, \phi) = \mathcal{S} \tag{2}$$

Given a finite set of trajectories, if $\mathcal{S}$, the object model is known, the optimal set of trajectories can be found with an exhaustive search. As the object model is not known apriori in an unknown environment, the optimal solution can not be found beforehand. Thus, each robot needs to move based on the partial observations of the team to determine the NBV to reconstruct the object's surface. Here we assume that each robot can observe the object at the start of the mission, which can be accomplished by moving the robots till they see the object. In this work, we define this problem in a centralized form; all the robots share their observations with a central entity that prescribes the NBV for each by solving the aforementioned objective.

[1]http://raaslab.org/projects/MAPNBV/

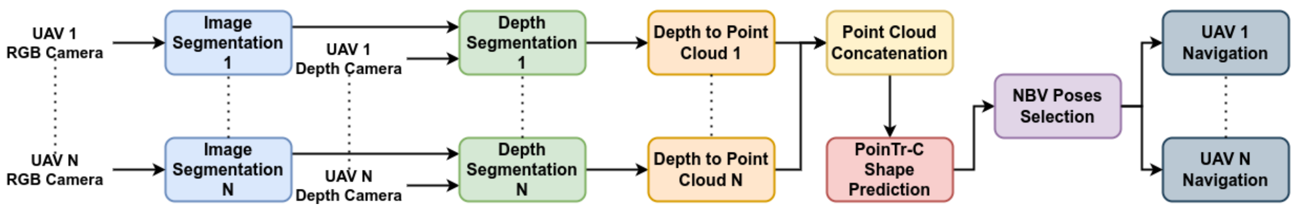


Fig. 2: Algorithm Overview

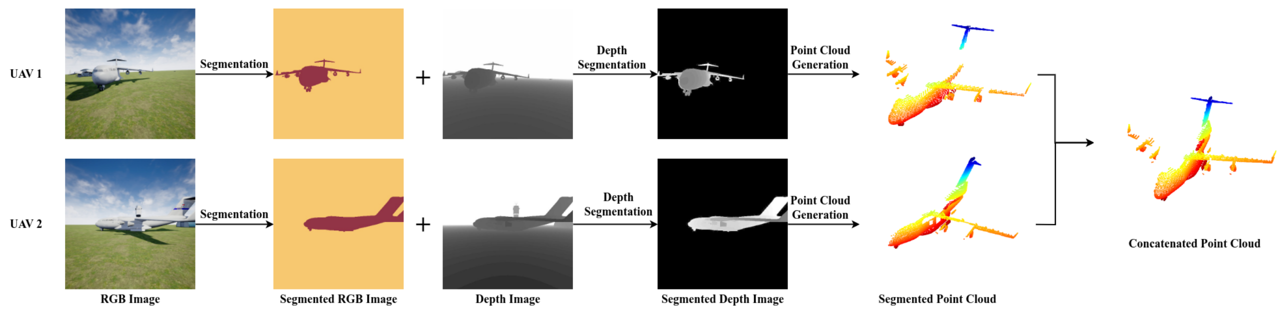


Fig. 3: Example observation point cloud generation and concatenation pipeline.

IV. Proposed Approach

In this paper, we present *Multi-Agent Pred-NBV (MAP-NBV)*, a model prediction-guided NBV approach for a team of robots. Figure 2 shows the overview of our process, which consists of two parts: (1)*3D Model Prediction*, where we combine the observations from all the robots to build a partial model of the object and use PoinTr-C [10], a 3D point cloud completion network, to predict the full shape of the objects, and (2) *Multi-Agent NBV Algorithm*, which uses the partial model and the predicted model to determine the NBV *for the team*, while trying to minimize the distance traveled. Our NBV solution performs a greedy selection over the candidate points to generate the trajectory, which also reduces the computation complexity. The following subsections provide further details of our approach.

A. 3D Model Prediction

To start, the target object is segmented out from the rest of the environment in the captured RGB images for each UAV. This allows the algorithm to focus on only the target infrastructure as opposed to also including other obstacles. Then, each of these segmented images is aligned with the captured depth image per UAV to segment the target object out. Point clouds are then generated per each segmented depth image. This gives us a point cloud per each UAV that contains points belonging only to the target object. Assuming a centralized system, each segmented point cloud per UAV is transformed into a central reference frame and concatenated together into a singular point cloud. This point cloud represents the entire multi-agent system's observations of the target object at the current timestamp. The point cloud concatenation can be replaced with a registration algorithm [33], but we use concatenation due to its ease of use. Lastly, this current timestamp's point cloud is then concatenated with previous observations to get an up-to-date observation point cloud. This process is shown in Figure 3.

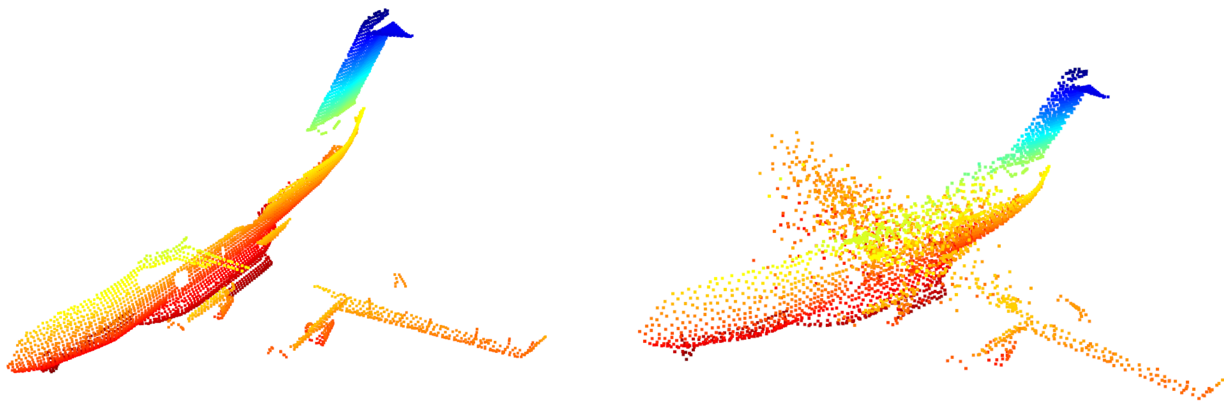
In order to get an approximation of the $\hat{\mathcal{V}}$ of the full model $\mathcal{V}$, we use PoinTr-C [10] a 3D point cloud completion network, developed by fine-tuning PoinTr [34] using curriculum learning over ShapeNet dataset [35]. Unlike PoinTr and similar point cloud completion networks, PoinTr-C doesn't make implicit assumptions about the knowledge of the center of the full model by fine-tuning over rotationally and translationally perturbed point clouds. Relaxing this assumption makes PoinTr-C more suitable for inputs from an unknown environment than PoinTr. The 3D point cloud of the object obtained as the union of the observed surface points goes as input to PoinTr-C and it predicts the full object point cloud $\hat{\mathcal{V}}$.

PoinTr-C was trained over isolated point clouds and therefore requires object point clouds to be isolated from the scene. This can be realized with the help of distance-based filters and state-of-the-art segmentation networks [36] without any fine-tuning. An example of an input point cloud and a predicted point cloud is shown in Figure 4.

B. Next-Best View Planner

We use the predicted point cloud as an approximation of the ground truth point cloud for NBV planning. For this, we first generate a set of candidate poses around the partially observed object. From these, we select a set of n poses, corresponding to each robot, based on information gain and control effort. The information gain for the set of n viewpoints is defined as the number of new, unique points expected to be observed after the robots move to these viewpoints. The control effort is defined as the total distance covered by the robots in moving to the viewpoints.

The number of new points varies in each iteration since the robots observe more of the surface of the object as they move to new locations. While PoinTr-C predicts the point cloud for the whole object, the robots can observe only the surface points. Hence, before counting the number of new points,



(a) Input (b) PoinTr-C Prediction

Fig. 4: Predicted point cloud on C-17 plane after initial observation.

we apply hidden point removal [37] to the predicted point cloud. We represent this relationship between the number of points observed and the trajectories traversed till time t by $I(\{\bar{\xi}_t\})$, where $\bar{\xi}_t = \{\xi_{r_1}, \xi_{r_2}, ..., \xi_{r_n}\}_t$ represents the set of trajectories for all the robots till time t. To balance the information gain and control effort, we use a hyperparameter τ which is kept fixed throughout an episode. The robots select the candidate to pose set which results in at least $\tau\%$ of the total possible information gain over all candidate poses. Thus, we formulate our multi-agent NBV objective as follows.

$$\{\phi_{r_1}, \phi_{r_2}, ..., \phi_{r_n}\}_{t+1} = \arg\min_{\phi \in \mathcal{C}} \sum_{i=1}^{n} d(\phi_{r_i}, \phi_{r_i t})$$
$$\textit{such that} \quad \frac{\bigcup_{i=1}^{n} I(\xi_{r_i t} \cup \phi)}{\max_{\phi \in \mathcal{C}} \bigcup_{i=1}^{n} I(\xi_{r_i t} \cup \phi)} \geq \tau$$

In our experiments, we implement the information gain by first isolating the predicted points that can be observed from a given set of viewpoints and then taking a union of such points from each agent to identify the unique points in the joint observation. The number of the points thus obtained is used as the information gain. For finding the control effort, we use RRT-Connect [38] to find the path between a robot's current location to each candidate pose. The candidate poses are generated similar to Pred-NBV [10], i.e. on circles at different heights around the center of the predicted object point cloud. One circle is at the same height as the predicted object center with radius $1.5 \times d_{max}$, where d_{max} is the maximum distance of a point from the center of the predicted point cloud. The other two circles are located above and below this circle $0.25 \times$ z-range away, with a radius of $1.2 \times d_{max}$. The viewpoints are located at steps of 30° on each circle. We set $\tau = 0.95$ for all our experiments.

V. Experiments and Evaluation

In order to gauge our method's effectiveness, we compare it with a non-predictive multi-agent baseline and a prediction-driven NBV approach which was developed for a single agent. While the first highlights the benefits of including predictions in the NBV pipeline, the latter supports the argument for using a team of robots.

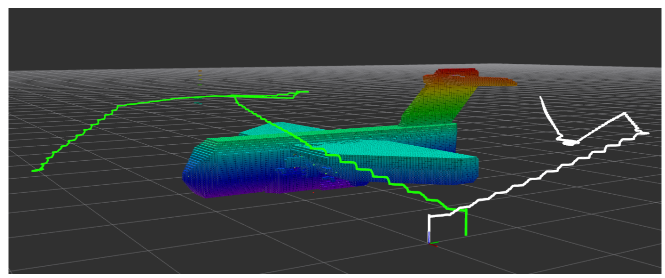


Fig. 5: Flight paths of the 2 drones during C-17 simulation. The white line is UAV 1's path. The green line is UAV 2's path.

A. Setup

We extend the setup in Pred-NBV [10] to work in a multi-agent setting. Similarly, we use Robot Operating System (ROS) Melodic and AirSim [39] on Ubuntu 18.04 for our simulation experiments. Multiple UAVs are spawned into the AirSim environment. We equipped each of the UAVs with a depth camera and an RGB camera. Each UAV published a segmented image using AirSim's built-in segmentation. We adapted the depth segmentation package from Pred-NBV to work with multiple UAVs. We then converted these segmented depth images into 3D point clouds. For our collision-free point-to-point planner, we use the MoveIt [40] software package implementing the work done by Köse [41].

B. Qualitative Example

We evaluate *MAP-NBV* on the same 20 objects that were used in Pred-NBV to allow a direct comparison. The 20 objects consist of 5 different ShapeNet classes: airplane, rocket, tower, train, and watercraft. Examples of each class are shown in Figure 6. These classes represent diverse shapes and infrastructures that are regularly inspected. Figure 5

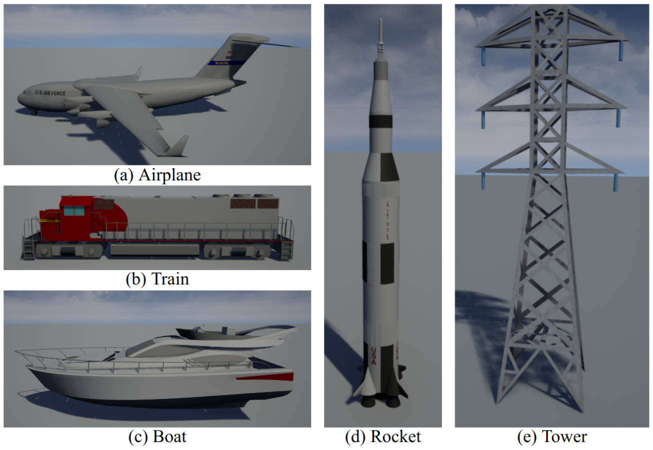


Fig. 6: Examples of the 5 simulation model classes.

shows the path followed by 2 UAVs as given by *MAP-NBV* in the C-17 airplane simulation. This environment includes other obstacles that are not of interest but still need to be accounted for in collision-free path planning. *MAP-NBV* finds a collision-free path for both UAVs while targeting the maximum coverage of the C-17 airplane.

C. Comparison with Single-agent Baseline

We compared the performance of *MAP-NBV* with a single-agent prediction-based NBV planner called Pred-NBV [10]. *MAP-NBV* is an extension of Pred-NBV designed for multi-agent scenarios. However, in single-agent cases, both algorithms function identically. In *MAP-NBV*, UAVs are spawned close together, ensuring that the initial environment information is virtually the same as in the single-agent Pred-NBV case. Consequently, the initial points observed and the initial shape completion predictions for both algorithms are highly similar. This means that *MAP-NBV* and Pred-NBV select their initial NBVs using nearly identical information.

To demonstrate the immediate information gain of *MAP-NBV* over Pred-NBV, we compare the number of points observed after navigating to the first NBVs selected by the algorithms. Our findings, presented in Table I, reveal that, on average, *MAP-NBV* observes 22.75% more points after the first iteration compared to Pred-NBV in the context of object reconstruction. These results are based on evaluations across 20 objects and 5 object classes.

Furthermore, on average, each UAV in *MAP-NBV* flew a similar distance to the UAV in Pred-NBV. This similarity arises from both algorithms generating candidate viewpoints in the same manner and employing the same point-to-point planner.

D. Comparison with Multi-agent Baseline

We also compared the performance of *MAP-NBV* with a modified baseline NBV method [11] designed for multi-agent use. The baseline method employs frontiers to select the next-best views. Frontiers are points located at the edge of the observed space near unknown areas. We utilized the same modifications described in Pred-NBV [10]. Specifically, we used our segmented point cloud to choose frontiers near the target object. To ensure that the UAVs always face the target object, the orientation of all poses selected by the baseline aligns with the center of the observed target object point clouds.

TABLE I: Points observed by *MAP-NBV* and Pred-NBV [10] after first iteration of algorithm.

Class	Model	Points Seen *MAP-NBV*	Points Seen Pred-NBV	Improvement
Airplane	747	**11145**	8570	26.12%
	A340	**6902**	5367	25.02%
	C-17	**10207**	7258	33.77%
	C-130	**5201**	2559	68.09%
	Fokker 100	**9941**	6843	36.92%
Rocket	Atlas	**1644**	1468	11.31%
	Maverick	**2535**	2257	11.60%
	Saturn V	**943**	941	0.21%
	Sparrow	**1294**	1098	16.39%
	V2	**1093**	949	14.10%
Tower	Big Ben	**2612**	1980	27.53%
	Church	**6281**	4589	31.13%
	Clock	**2531**	1971	24.88%
	Pylon	**2772**	2600	6.40%
	Silo	**4188**	3168	27.73%
Train	Diesel	**3228**	3197	0.96%
	Mountain	**4243**	4174	1.64%
Watercraft	Cruise	**3359**	1686	66.32%
	Patrol	**3684**	3677	0.19%
	Yacht	**10114**	7892	24.68%

We further adapted this baseline method to function in a multi-agent setting. The pose for the first UAV is selected in the exact same manner as in the single-agent baseline. For each subsequent UAV, the remaining best pose is chosen, as long as it does not fall within a certain distance threshold compared to the previously selected poses in the current iteration of the algorithm.

Both *MAP-NBV* and the baseline algorithm employ the same stopping criteria. The algorithm terminates if the total points observed in the previous step exceed 95% of the total points observed in the current step. Our evaluation, presented in Table II, demonstrates that *MAP-NBV* observes, on average, 15.63% more points than the multi-agent baseline for object reconstruction across all 20 objects from the 5 different model classes. In our simulations, we utilized 2 UAVs for both algorithms.

Furthermore, the *MAP-NBV* algorithm can be readily extended to accommodate more than just 2 robots. By incorporating additional UAVs, the algorithm can effectively leverage the collaborative efforts of a larger multi-agent system to improve object reconstruction performance and exploration efficiency. However, in our current evaluation, we utilized 2 UAVs for both algorithms due to limited computational resources. The simulations were computationally intensive, and our computer experienced significant slowdowns with just 2 robots in the simulation. Despite this limitation, the promising results obtained with 2 UAVs suggest that scaling up the algorithm to include more robots has the potential to yield even more significant improvements in performance.

Additionally, Figure 7 illustrates that *MAP-NBV* observes more points per step than the multi-agent baseline while also covering a shorter flight distance.

TABLE II: Points observed by *MAP-NBV* and the multi-agent baseline NBV method [11] for all models in AirSim upon algorithm termination.

Class	Model	Points Seen *MAP-NBV*	Points Seen MA Baseline	Improvement
Airplane	747	**11628**	10214	12.95%
	A340	**9202**	8156	12.05%
	C-17	**12599**	10150	21.53%
	C-130	**6311**	5961	5.70%
	Fokker 100	**15613**	13158	17.07%
Rocket	Atlas	**1879**	1747	7.28%
	Maverick	**3357**	2693	21.95%
	Saturn V	**985**	877	11.60%
	Sparrow	**1797**	1664	7.69%
	V2	**1255**	919	30.91%
Tower	Big Ben	**3741**	3493	6.86%
	Church	**8004**	6890	14.96%
	Clock	**3139**	2382	27.42%
	Pylon	**3075**	2870	6.90%
	Silo	**5933**	4296	32.01%
Train	Diesel	**3427**	3233	5.83%
	Mountain	**4711**	4215	11.11%
Watercraft	Cruise	**4746**	3118	41.40%
	Patrol	**3989**	3683	7.98%
	Yacht	**11351**	10341	9.31%

VI. Conclusion

We present a multi-agent, prediction-guided NBV planning approach for active 3D reconstruction. This method can be helpful in a variety of applications including civil infrastructure inspection. We show that our method is able to faithfully reconstruct the object point clouds efficiently compared to non-predictive multi-agent methods and single-agent prediction-based methods. Our NBV planning objective considers both information gain and control effort, making it more suitable for real-world deployment given the flight time limit imposed on UAVs by their battery capacity.

In this work, we focus solely on geometric measures for information gain. Many existing works on NBV have developed sophisticated information theoretic measures. We will explore combining both types of measures in our future work. Also, we consider all possible viewpoint pairs for finding the NBV for the team, which hinders the scalability of MAP-NBV. We will look into methods to make this process more computationally efficient search over a larger candidate viewpoint set.

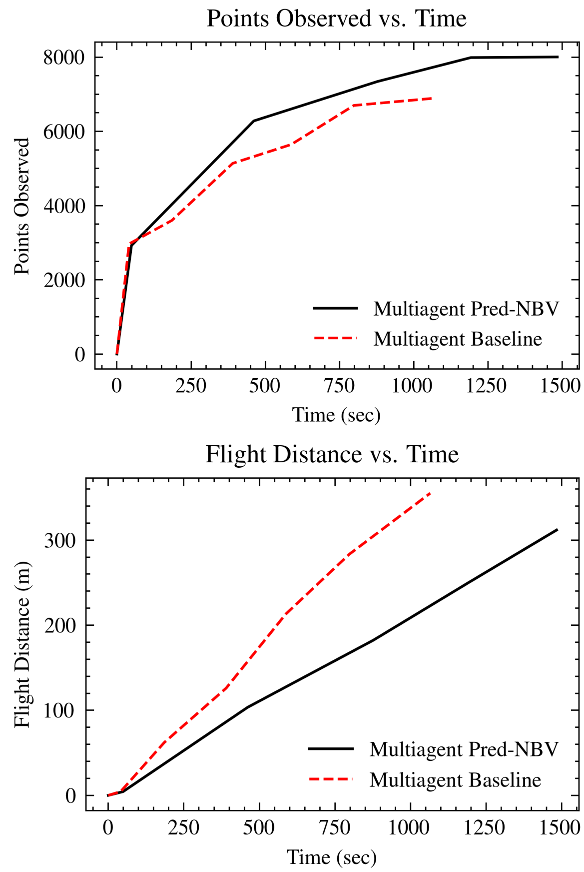


Fig. 7: Comparison between **MAP-NBV** and the **Multi-agent baseline NBV algorithm** [11] for the Church Tower.

References

[1] P. Shanthakumar, K. Yu, M. Singh, J. Orevillo, E. Bianchi, M. Hebdon, and P. Tokekar, "View planning and navigation algorithms for autonomous bridge inspection with uavs," in *International Symposium on Experimental Robotics (ISER)*, 2018, accepted.

[2] H. Dhami, K. Yu, T. Williams, V. Vajipey, and P. Tokekar, "GATSBI: An online GTSP-based algorithm for targeted surface bridge inspection," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2023. [Online]. Available: https://doi.org/10.1109%2Ficuas57906.2023.10156013

[3] A. Kim and R. Eustice, "Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1559–1565.

[4] L. Ropek, "This startup is using drones to conduct aircraft inspections," Mar 2021. [Online]. Available: https://gizmodo.com/this-startup-is-using-drones-to-conduct-airplane-inspec-1846527002

[5] H. Dhami, K. Yu, T. Xu, Q. Zhu, K. Dhakal, J. Friel, S. Li, and P. Tokekar, "Crop height and plot estimation for phenotyping from unmanned aerial vehicles using 3d lidar," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[6] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 37, 2018. [Online]. Available: https://arxiv.org/abs/1805.09817

[7] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=-v4OuqNs5P

[8] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, pp. 177–196, 2018.

[9] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3d object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[10] H. Dhami, V. D. Sharma, and P. Tokekar, "Pred-nbv: Prediction-guided next-best-view for 3d object reconstruction," 2023.

[11] J. Aleotti, D. L. Rizzini, R. Monica, and S. Caselli, "Global registration of mid-range 3d observations and short range next best views," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3668–3675.

[12] T. Ozaslan, S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Inspection of penstocks and featureless tunnel-like environments using micro uavs," in *International Conference on Field and Service Robotics*, 2013.

[13] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24 –39, Mar 2012.

[14] Y. Sung and P. Tokekar, "A competitive algorithm for online multi-robot exploration of a translating plume," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2019.

[15] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1378–1385.

[16] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and autonomous systems*, vol. 8, no. 1-2, pp. 47–63, 1991.

[17] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3d object reconstruction with positioning error," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 159, 2014.

[18] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[19] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[20] B. Adler, J. Xiao, and J. Zhang, "Autonomous exploration of urban environments using unmanned aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 6, pp. 912–939, 2014.

[21] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[22] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.

[23] J. E. Banta, L. Wong, C. Dumont, and M. A. Abidi, "A next-best-view system for autonomous 3-d object reconstruction," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 5, pp. 589–598, 2000.

[24] S. Kriegel, M. Brucker, Z.-C. Marton, T. Bodenmüller, and M. Suppa, "Combining object modeling and recognition for active scene exploration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2384–2391.

[25] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3813–3822.

[26] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta, "Supervised learning of the next-best-view for 3d object reconstruction," *Pattern Recognition Letters*, vol. 133, pp. 224–231, 2020.

[27] R. Zeng, W. Zhao, and Y.-J. 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)*. IEEE, 2020, pp. 7050–7057.

[28] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[29] A. A. Amanatiadis, S. A. Chatzichristofis, K. Charalampous, L. Doitsidis, E. B. Kosmatopoulos, P. Tsalides, A. Gasteratos, and S. I. Roumeliotis, "A multi-objective exploration strategy for mobile robots under operational constraints," *IEEE Access*, vol. 1, pp. 691–702, 2013.

[30] G. Hardouin, J. Moras, F. Morbidi, J. Marzat, and E. M. Mouaddib, "Next-best-view planning for surface reconstruction of large-scale 3d environments with multiple uavs," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1567–1574.

[31] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "Multi-robot hybrid coverage path planning for 3d reconstruction of large structures," *IEEE Access*, vol. 10, pp. 2037–2050, 2021.

[32] C. Wu, R. Zeng, J. Pan, C. C. Wang, and Y.-J. Liu, "Plant phenotyping by deep-learning-based planner for multi-robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3113–3120, 2019.

[33] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," 2021.

[34] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 498–12 507.

[35] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[36] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

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

[38] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[39] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[40] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[41] "tahsinkose/hector-moveit: Hector quadrotor with moveit! motion planning framework," https://github.com/tahsinkose/hector-moveit, (Accessed on 03/03/2021).