Learning the Next Best View for 3D Point Clouds via Topological Features

Christopher Collander, William J. Beksi, and Manfred Huber

***Abstract*— In this paper, we introduce a reinforcement learning approach utilizing a novel topology-based information gain metric for directing the next best view of a noisy 3D sensor. The metric combines the disjoint sections of an observed surface to focus on high-detail features such as holes and concave sections. Experimental results show that our approach can aid in establishing the placement of a robotic sensor to optimize the information provided by its streaming point cloud data. Furthermore, a labeled dataset of 3D objects, a CAD design for a custom robotic manipulator, and software for the transformation, union, and registration of point clouds has been publicly released to the research community.**

I. Introduction

Modern robots are equipped with a wide variety of sensors and actuators for observing and interacting with the surrounding environment. This allows a robot to carry out applications such as 3D reconstruction, object recognition, grasping, and much more. To receive further clues about its surroundings, a robot must move its sensor to another location and obtain new sensor values [1, 2, 3]. Several factors constrain the acquisition of the next sensor view including the kinematics of the robot, the field of view and range of the sensor, and environmental obstructions such as occlusions [4] and obstacles [5]. The challenge of obtaining a series of updated sensor placements is known as the next best view (NBV) problem. Specifically, given a known sensor pose and an information value, the next placement of the sensor should ensure that the value obtained maximizes the 'information gain' across the full action space of the sensor, Fig. 1. Information gain is calculated between two subsequent views, and the full gain of a sequence of views is the sum of the individual consecutive gains.

The definition of information gain is dependent upon the end goal of the robot. In the context of reconstructing the surface of an object, information gain may be defined as the metric of resolution [6], mesh quality [7], or volumetric representation [8]. For the task of object recognition, information gain can be formalized as the fusion of successive object hypotheses [9], a metric of prediction surety [10], or a combination of probabilistic and volumetric metrics [11]. In the case of picking and transporting objects by robots in potentially cluttered environments, information gain can be described by KL divergence [12], the utility of the sensor pose based on regions of interest [13], or the occupancy probability of a voxel up to the most recent observation [14]. Enabling robots that can act autonomously in dynamic, unstructured settings is a major challenge. In such circumstances, learning to reconstruct, recognize, and manipulate unfamiliar objects by choosing the NBV is an important capability.

The authors are with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA. Emails: christopher.collander@mavs.uta.edu, william.beksi@uta.edu, huber@cse.uta.edu.

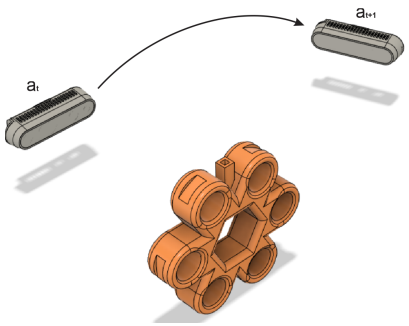

Fig. 1. The next best view (NBV) problem is the challenge of determining where to relocate a sensor, after an initial reading, to maximally increase the knowledge of the environment.

In contrast to the aforementioned work, we propose to define information gain in terms of persistent topological features modeled as rewards. The key insight of this effort is that the incorporation of topological attributes can yield unique knowledge with respect to the structure of point cloud data which is not obtainable from other metrics. Our work makes the following contributions: (i) a fully automated online deep reinforcement learning approach for achieving the NBV on streaming 3D point cloud data; (ii) a novel and generalizable topology-based information gain metric; (iii) the public release of a point cloud dataset for seven exclusive objects in various labeled positions, an open-source CAD design for a custom robot manipulator, and open-source software for the transformation, union, and registration of point clouds given their viewpoint coordinates.

The remainder of this paper is organized as follows. Related literature is reviewed in Sec. II and the background material for this work is provided in Sec. III. The statement of the problem is given in Sec. IV. Our reinforcement learning approach to directing the NBV via the computation of a topology-based information gain metric is presented in Sec. V. The design and results of our experiments are discussed and demonstrated in Sec. VI. In Sec. VII, the paper is concluded and future work is mentioned.

II. Related Work

Deciding the NBV is a fundamental area of research in active vision. It has over thirty years of history and remains an open problem [15]. We distinguish between model-based NBV approaches, where the views can be planned offline, and non-model-based algorithms where the NBV is selected at runtime since no a priori knowledge about the target object

is available. A comprehensive survey on model-based and non-model-based methods is provided by [16]. Our approach develops a non-model-based scheme for finding the NBV.

An automated surface acquisition system employed the NBV by partitioning a viewpoint into seen and unseen views [17, 18]. A list of desired constraints for future views is constructed to increase the expected quality and reduce the search state space. Referred to as the PS algorithm, this method relies on determining which views constrain the ranging rays to be colinear with observation rays of the object surface from the visual sensor. Similar to our research, this work prioritizes views on the edge of the initial view, resulting in some overlap between the views. In contrast to our work, multiple views are evaluated before determining which is the NBV. This method also only evaluates the NBV across a single dimension via the use of a turntable to manipulate the object.

To determine which sensor poses have the highest probability for viewing missing cells in a model, researchers have established a probabilistic framework for expected information gain [19]. In their work, a voxelized occupancy grid was used for calculations in non-simulated, real-world scenarios of multiple objects cluttered on a small table. A path planning algorithm utilizing rapidly-exploring random trees was proposed by researchers for an aerial vehicle with a stereo camera to determine the NBV [20]. To explore a 3D environment, their algorithm chooses views based on the amount of unmapped space. A voxelized occupancy grid was used in a simulated environment for testing, and successful real-world tests were also done with an unmanned aerial vehicle. Dissimilar to our work, both [19] and [20] attempt to determine the NBV of a full environment while our research focuses on the NBV of a single object.

In [21], point clouds from ShapeNet [22] objects were examined for the NBV with a novel surface coverage metric. Unlike our work, the research focuses on reconstructing an object in 3D space. In contrast, our research concentrates on maximizing the realization of features in the observed manifold. Specifically, we maximize the number of physical holes and minimize both the number of holes due to missing points and the number of connected components observed in the data. Another difference is that our work is based on a single future view. On the contrary, [21] takes multiple views while attempting to minimize how many views are required for reasonable surface coverage.

The research proposed in [23] utilizes deep Q-learning for multi-view planning with a reward value focused on depth-image inpainting. Different from our work, this research does not convert the depth map into a point cloud and therefore it does not calculate any graph-based values as we do. Additionally, [23] cannot find points outside of the dimensions of the original depth image nor in opposing viewpoints. In our research, the pose can be positioned in any location in spherical coordinates. [23] makes depth decisions of a full environment while our system works on the observation of a single segmented object in space.

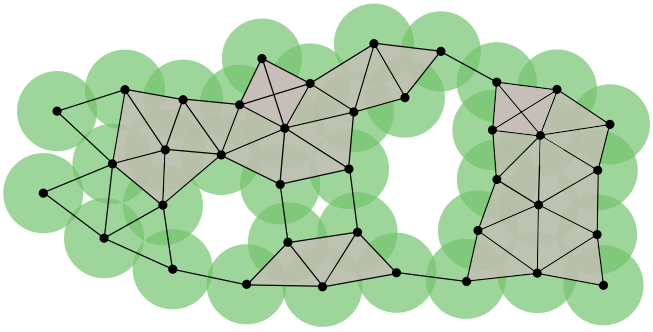

Fig. 2. 0-simplices (vertices), 1-simplices (edges), and 2-simplices (shaded triangles) can be glued together to form a Vietoris-Rips (VR) complex.

III. Background

This section provides the necessary background material on the computational aspects of topological data analysis [24] and reinforcement learning [25] upon which our work is based.

A. Topological Data Analysis

Topological data analysis (TDA) refers to a collection of tools for extracting topological features from data [26]. The main tool, persistent homology, allows us to study homology (i.e., connected components, holes, and higher dimensional analogs) at multiple scales [27]. Concretely, it provides a basis to quantify the evolution of the homology of a parameterized family of topological spaces.

1) Simplicial Complex Representation: The discrete space that we work in uses simplices as its building blocks. A k-dimensional simplex σ is the convex hull of $k+1$ affinely independent vertices $v_0, \ldots, v_k \in \mathbb{R}^n$. For example, 0-simplices are vertices, 1-simplices are edges, and 2-simplices correspond to triangles. Given a set of points $\{x_0, \ldots, x_{m-1}\} \in \mathbb{R}^n$ and fixed radius r, the Vietoris-Rips (VR) complex is an abstract simplicial complex (i.e., collection of simplices closed under the operation of taking subsets) such that

$$K = \{\sigma \subset \{x_0, \ldots, x_{m-1}\} \mid dist(x_i, x_j) \leq r, \forall x_i \neq x_j \in \sigma\}.$$

In this definition, $dist$ is the Euclidean distance function and the points of σ are pairwise within distance r of each other, Fig. 2.

2) Chains, Boundaries, and Cycles: Given an abstract simplicial complex K, a k-chain is a subset of k-simplices in K. A boundary, $\partial_k(\sigma)$, is a collection of $(k-1)$ dimensional simplices and forms a $(k-1)$-chain. Taking the sum of the boundaries of the simplices in a k-chain gives the boundary of the k-chain, $\partial_k(c) = \sum_{\sigma \in c} \partial_k(\sigma)$. A homomorphism, $\partial_k : C_k \to C_{k-1}$, is defined for each boundary operator and the collection of boundary operators on the chain groups forms a chain complex,

$$\emptyset \to C_3 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \emptyset.$$

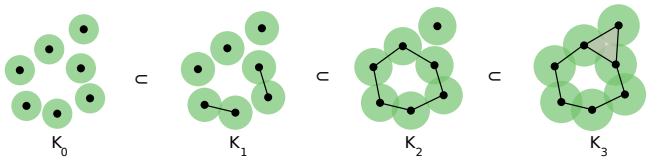

Fig. 3. A growing sequence of subcomplexes is known as a filtration.

3) Homology Groups: The homology groups comprise vector spaces where a subset yields a vector space if every element is the sum of elements in the subset. A minimal generating set serves as a basis for the space. The rank of the kth homology group is defined as the kth Betti number of K, e.g. $\beta_k = \text{rank } H_k$ where

$$\text{rank } H_k = \text{rank } Z_k - \text{rank } B_k.$$

Only the Betti numbers for $0 \leq k \leq 2$ can be non-zero for complexes in $\mathbb{R}^3$. A non-bounding 0-cycle represents a set of components of K where there is one basis element per component. Thus, β_0 is the number of components that make up K. A set of holes formed by K is represented by a non-bounding 1-cycle. Hence, each hole can be expressed as a sum of holes in a basis and β_1 is the size of the basis.

4) Filtrations: The evolutional growth of a complex of an increasing sequence of topological spaces is called a filtration, Fig. 3. Concretely, if $f : X \to \mathbb{R}$ is defined on a topological space X then each sublevel set $X_r = f^{-1}((-\infty, r])$ generates a topological space X_r where $X_r \subset X'_r$ and $r \leq r'$. As the spatial neighborhood r increases, homological features are born (created) and can die (disappear). In this work, we use filtrations of finite simplicial complexes in $\mathbb{R}^3$,

$$\emptyset \subset K_0 \subset K_1 \subset \ldots \subset K_L = K_\infty,$$

where L is a maximum threshold value for constructing a complex.

5) Persistent Homology: Persistence measures the lifetime of topological attributes in a filtration. More precisely, given the lth complex K^l in a filtration, let Z^l_k and B^l_k be the corresponding kth cycle group and kth boundary group, respectively. Then the persistent cycles in K^l are obtained by factoring the kth cycle group by the kth boundary group of K^{l+p}, p complexes later in the filtration. The p-persistent kth homology group of K^l is defined as

$$H^{l,p}_k = Z^l_k \,/\, (B^{l+p}_k \cap Z^l_k),$$

and the rank of $H^{l,p}_k$ is the p-persistent kth Betti number $\beta^{l,p}_k$ of K^l.

B. Reinforcement Learning

Reinforcement learning (RL) is an iterative process between an agent and the environment. The agent receives information about the environment, along with a reward for the previous action, and then decides on a new action that will affect the environment. Over time, the agent learns how to perform actions that will result in the largest cumulative reward.

The action space, $\mathcal{A}$, is the set of all possible actions, a, that can be made by an agent ($a \in \mathcal{A}$). The observation space, $\mathcal{S}$, consists of information concerning the state s of the environment that can be used by an agent in deciding what action to take ($s \in \mathcal{S}$). A transition function, $T(s, a, s')$, is the probability that action a from state s leads to state s'. The output of a reward function, $R(s, a, s')$, is represented by a real number and allows for deciding an action based on a decision, which may or may not move an agent closer to a goal. A policy, π, is a function that makes a probabilistic decision on what action to take given an observation and previously known information about the environment. Formally,

$$\pi \colon \mathcal{A} \times \mathcal{S} \to \mathbb{R}, \quad \pi(a, s) = p(a \,|\, s).$$

Many RL situations are sequential, i.e., more than a single action-reward pair is iteratively performed. In contrast, scenarios in which only a single action is taken and then the final reward is obtained are called multi-armed bandits. Since this research explores the NBV using just a single view, it can be modeled as a multi-armed bandit problem.

IV. Problem Statement

Let $X = \{x_0, \ldots, x_{m-1}\} \in \mathbb{R}^3$ be a topological space where $x_0, \ldots, x_{m-1}$ are the points in a point cloud captured by a 3D sensor. Our objective is to develop an online reinforcement learning methodology utilizing persistent topological features for acquiring the NBV. To accomplish this, we first approximate the topology of the space through a VR complex. Next, we compute the view's information based on the Betti numbers of the point cloud over a range of spatial resolutions and a deep reinforcement learning action-value network is designed and trained using the output of our topological information gain metric. The network is used to evaluate the NBV on a test set of 3D objects, and the results are analyzed and discussed.

V. Approach

To find the NBV, we interpret RL actions as sensor poses. More specifically, we model the action space as the set of all possible poses in $\mathbb{R}^6$. Since a 3D point cloud is an unordered set of points, feeding it directly into the network does not provide any meaningful information on its own. Instead, we interpret the observation space as the Betti numbers of the topological filtration of the initial view's point cloud. We obtain the information gain for the NBV as follows. First, a topological filtration is performed by building a VR complex at multiple spatial resolutions on the input point cloud. Next, the Betti numbers calculated for each resolution are algebraically combined to provide a single value that represents topological characteristics in the observed manifold of the object.

Intuitively, when observing a singular object *minimizing* β_0 will result in an improved view by ensuring that disjoint sections become connected through an additional view. This also minimizes the likelihood of incorporating prior views. These prior views can have higher levels of noise outside of

the observed object thus causing the values of β_0 to increase. Conversely, *maximizing* β_1 will lead to an increase in high-detailed sections of the object manifold such as concave sections, corners, and holes.

Due to feature scaling and the irregular density of a point cloud, a single (β_0, β_1) pair may not always provide maximal information. For this reason, a filtration records multiple (β_0, β_1) pairs at an increasing range of distances. Thus, we define the value of a view with topological attributes obtained at distance r as

$$V = \alpha \sum_{r \in D} \beta_1(r) - (1 - \alpha) \sum_{r \in D} \beta_0(r), \tag{1}$$

where D is the set of all neighborhood point radii. The coefficient $\alpha \in [0, 1]$ can be adjusted as necessary based on the importance of connected components (β_0) versus holes (β_1). Given a point cloud P_t acquired at time step t, we define the reward for the NBV at t as

$$R_t = V(P_{t-1} \cup P_t) - V(P_{t-1}). \tag{2}$$

This differential reward metric attempts to minimize the number of connected components and the number of noisy areas with missing data while maximizing the number of legitimate physical holes observed in the manifold.

VI. Experiments

In the following subsections, we present our experimental setup and results for computing the NBV as described in Sec. V. All experiments were conducted using streaming 3D sensor data and a robotic manipulator. Our software, data, and models have been made publicly available to the research community [28].

A. Experimental Design

To determine the NBV, a sensor must observe an object from an initial view and then decide another view based on the computed value of the information gain. The possible positions of a sensor around an object are 6-dimensional, i.e., translation in x, y, z and rotation in θ, ϕ, ψ. This dimensionality may be reduced by constraining the position of the sensor relative to the observed object such that it always points directly towards the object, is located parallel to the ground plane, and is translated a specific distance away from the center of the object. By making these adjustments, the pose space can be reduced to two variables: pitch and yaw. In a simulated environment, we would be able to position the sensor around an object as necessary. However, sensor positioning constraints in the real world constitute many complications. To overcome these challenges, a 2-DOF manipulator was created to facilitate the data capture. 3D point cloud data was collected with an Intel RealSense D415 stereo camera.

The central sensor line was made coincident with the origin of the object of observation. Two motors (orthogonal to the sensor) were placed with axes coincident to the central origin of the object of observation. This allowed the bottom motor to control the pitch and the top motor to control the yaw of the object.

The action space is all possible positions of the motors. For the manipulator, a total of 4,096 positions are possible for each motor. Since the pitch of the motor is not able to move a full 360° due to the mechanics of the manipulator, we restricted the pitch value to 2,048 positions. Although this helped reduce the action space, it was still too large for any efficient RL method. Therefore, we discretized the action space into buckets to further reduce the number of possible actions to a reasonable value. For the experiments, 20 yaw buckets and 10 pitch buckets were decided which resulted in a final action count of 200.

A minimum of 5,000 views were randomly sampled uniformly from the full action space of the motors for each object. These views were saved along with a record of the motor positions for each view. The point cloud RGB values were not used in the experiments. We performed the data collection process for seven different objects. Three of these items were CAD generated and included many concave sections and holes. Three other objects were constructed from various household objects and a toy View-Master was used for testing the system, but not for training. Public datasets such as ShapeNet do not include data holes, noise, or missing sections from camera angles, therefore our score metric for the NBV would be inaccurate without adjustments to the data.

Transform/Union/Registration (TUR) is a software application written for our experiments to combine point clouds. Given multiple point clouds with yaw/pitch values, TUR will iteratively process each dataset. The first step for each point cloud is to transform the points into the coordinate system of the world frame. Next, the iterative closest point algorithm [29] is used for registration by matching the points in the world frame with all earlier obtained point clouds under the presence of noise or manipulator bending. The last step is to union the current point cloud to all previously processed point clouds. TUR outputs a single point cloud file with all of the previous view information. Due to the noise from the sensor and environment, as well as bending in the manipulator, the unions are not always perfect. However, we observed that similar reward values were obtained despite these issues.

For a trainable RL policy, the simplest implementation is a value-averaging tabular approach. In this method, a table entry is required for every action and observation pair. However, while the action space is effectively discretized, the observation space is surprisingly large. We observed values of β_0 and β_1 greater than 200 in many samples. Based on our initial observations, we designed a dense multilayer neural network for deep reinforcement learning. The network takes the role of the tabular system with observations as input and values as output, i.e., one value for each action. These values are then used to decide what action to take. The advantage is that the observation space, given as input to the neural network, does not need to be discretized.

We built a VR complex over the registered point cloud

data via a multistep filtration to obtain a set of (β_0, β_1) pairs using the approach described in [30].

Since the coordinates of the object are in an absolute world frame, the action taken by the NBV is highly dependent on the previously observed location. Therefore, the initial yaw and pitch values need to be known to determine the NBV. For this reason, we include the starting yaw and pitch in the observation space in addition to the pairs of Betti numbers.

We note that for the observations of unioned views, the cardinality of the space of unioned-views can be further reduced. For example, when $V_{t+1} = V_t$ the information gain is zero and the Betti numbers remain the same (i.e., the point cloud remains unchanged). Moreover, the union operation is commutative. Thus, our approach identifies these duplicate cases and greatly reduces the space of unioned-views from 200×200 to $\binom{200}{2}$ = 19,900.

B. Experimental Results

A neural network was trained with 2 dense layers of 64 nodes and 128 nodes respectively, with a leaky ReLU activation function [31]. The output layer consisted of 200 nodes with a linear activation function. Dropout, batch normalization, and L_2 regularization were utilized to prevent overfitting and object-remembrance. To calculate the Betti numbers and the value of a view (1), our model used a three-step filtration with the following radii: 0.002, 0.003, 0.004. These radii were chosen from observations of the point cloud density given the camera model and distance from the object. Based on our observations of consistently higher values for β_1 compared to β_0, an α of 0.15 was chosen to give the features approximately equal weight.

The training was completed in batches of 64 samples where the initial view and object were randomly sampled. The initial observations were referenced from the generated hash table. All future possible actions were then obtained by the union of the observations. The network was trained with the reward labels (2) across all output action nodes. After 512 batches of random initial views and objects, a test run of 512 batches was completed using $\epsilon = 0$. The mean reward for the test batch was recorded and this train/test cycle was completed 1,000 times.

By visualizing how the rewards are distributed we can gain a better understanding of the training process. To do this, we graph the action space of the view in two dimensions and plot the predicted NBV reward values of the trained model as a heatmap. We then visualize the heatmap of the final trained model on each object with different starting views, Fig. 4 (a)-(d). The initial views are indicated by blue squares in each plot and the NBVs (i.e., $\arg\max_a R(s,a)$) are illustrated by green circles. Lighter colors in the heatmap indicate higher reward values, representing views that are predicted to provide a large information gain.

From Fig. 4 we see that reward values cluster into regions of both optimal and negative views. For example, for the pair $(yaw = 17, pitch = 3)$, 42% of the test cases consider this point to be the NBV. Pitch values in the interval $[0, 2]$ consistently show negative rewards, as does a pitch value of

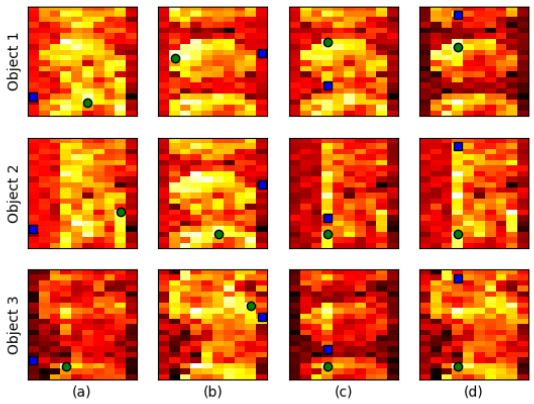

Fig. 4. Examples of action-value heatmaps with $(Y = yaw \in [0, 19], X = pitch \in [0, 9])$ for the three CAD objects, and four (a)-(d) random beginning views. The blue squares represent the initial views while the green circles are the NBV. Lighter colors in the heatmap correspond to higher reward values.

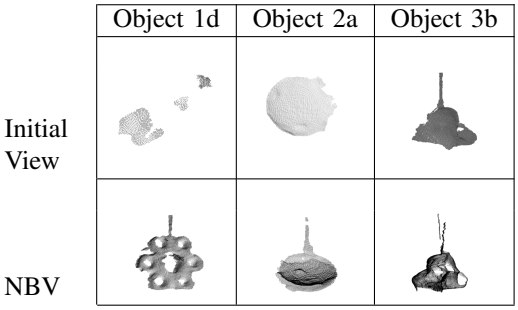

Fig. 5. Examples of initial views and the NBVs for the objects in Fig. 4.

9. On the other hand, a pitch value of 3 shows a light-colored column in most of the object heatmaps thus considering it to be a good view. Repetitive regions can be seen in the heatmaps for object 1(a) and 1(b) in Fig. 4. These repeating areas are also present in the heatmap of object 3(b) and 3(c), albeit slightly shifted. The shift can be explained as the NBV is contingent on relative viewing angles. Note that the yaw coordinate is a full 360°, i.e., each image wraps around vertically. Two primary reasons for these regions are the following: since we consider angles that are relative to the object frame each of the objects has a 90° and 180° symmetry, and certain positions of the manipulator occlude the view in many cases.

The point clouds of objects 1(d), 2(a), and 3(b) are shown in Fig. 5. For object 1(d), the initial view provides little detail

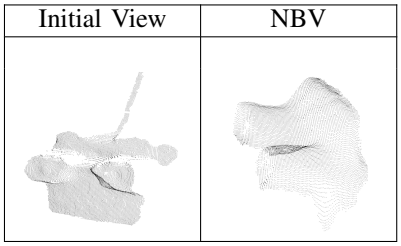

Fig. 6. An example of an initial view and the NBV for the test object.

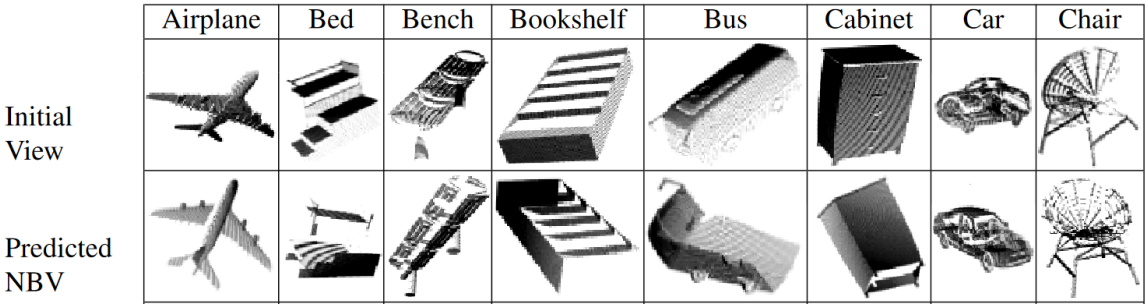

Fig. 7. Examples of initial views and the NBVs for various ShapeNet [22] objects.

and few points due to the obstruction of the manipulator. The predicted NBV is a partially-sideways front view that allows the two disjoint sections to join together into a single cluster when merged with the original pieces. For object 2(a), the spherical bottom of the object is initially observed. The prediction for the NBV is a side view of the object, and a larger representation of the object is realized when the union is performed. The inconsistency of the transform can be attributed to the bending of the manipulator. For object 3(b), the initial view is a full side view. The NBV prediction is of a partial side view of the object. Given these two views, there is enough overlapping information regarding the sides of the object in each side view that they can be unioned as one view. For all these examples, the NBV provided additional viewpoints that allowed intersecting information between the initial and ensuing views. This resulted in cohesive viewpoints of increasing detail. It shows that our topological information gain metric and RL approach correctly identify high-reward views, and it effectively increases the point cloud density around probable features.

Since the previous tests were completed using training data, a seventh real-world object was used for unbiased testing afterwards. The information gain values were calculated from the initial and union Betti numbers as with the previous objects, but these values were not used when training the RL model. The NBV returned from our RL model on an example initial view can be seen in Fig. 6. In this example, the initial view is rather good. Yet, there is an entire missing section on the back side of the object in addition to holes underneath and above the eye section due to being coplanar with the camera. In the predicted NBV, a lower angle is obtained which allows many of the missing sections to be cleanly filled and provides finer detail around the edges and curves.

To further test the model after training, 16 ShapeNet objects from the same categories as in [21] were placed into a simulated Gazebo [32] environment. A Robot Operating System [33] model for the simulated manipulator was created and a virtual Intel RealSense camera with parameters matching the physical model was used. With this system, the data collection process was repeated with the ShapeNet objects and the NBV was predicted using the trained model. As shown in Fig. 7, the trained NBV model tends to favor viewpoints that will partially overlap the initial viewpoint. This creates a combined manifold that attempts to better cover feature-filled areas of the objects such as holes and edges. In Fig. 7, the initial views were randomly chosen in a uniform manner. In addition, the predicted NBV images consist of only the second view's point cloud, as opposed to the unioned view. This allows for an easier visualization of the views.

To better quantify the decisions of the NBV on the ShapeNet objects, ten initial views from each object were evaluated with our trained model. The resulting angular difference between the initial view and the NBV were calculated. Of the 160 tests, the mean angular difference was $61.8°$ with a standard deviation of $19.2°$. These results fit with the intuitive beliefs of our metric. Very small angles will not correctly fill in holes of missing points due to occlusions. Large angles (e.g., $180°$) tend to increase the number of connected components and sections with missing points resulting in a lower value from our metric. Thus, viewpoints are chosen that are neither too far nor too close to the initial view which allows for probable feature-filled sections to be better realized.

VII. Conclusion

This paper introduced an online RL framework, modeled as a multi-armed bandit problem, for streaming 3D point cloud data. Moreover, we proposed a unique topology-based information gain metric to determine the NBV for a noisy 3D sensor. The metric focuses on topological features, such as holes and concave sections, and computes information gain by combining disjoint sensor viewpoints. Experimental results show that our system can help determine the placement of an object held by a robotic manipulator. As part of this work, a labeled dataset of 3D objects, a CAD design for a custom robotic manipulator, and software for the transformation, union, and registration of 3D point clouds given their viewpoint coordinates have been publicly released.

Acknowledgments

This material is based upon work supported by the National Science Foundation through grants #IIS-1724248 and #IIS-1947851, and a Department of Education's Graduate Assistance in Areas of National Need grant #1261801280. We thank Joseph M. Cloud for helpful discussions and comments on this work.

REFERENCES

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[2] J. Aloimonos, "Purposive and qualitative active vision," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 1. IEEE, 1990, pp. 346–360.

[3] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[4] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 15, no. 5, pp. 417–433, 1993.

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

[6] A. Hilton and J. Illingworth, "Geometric fusion for a hand-held 3d sensor," *Machine Vision and Applications*, vol. 12, no. 1, pp. 44–51, 2000.

[7] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.

[8] R. Monica and J. Aleotti, "Surfel-based next best view planning," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3324–3331, 2018.

[9] L. Paletta and A. Pinz, "Active object recognition by view integration and reinforcement learning," *Robotics and Autonomous Systems*, vol. 31, no. 1-2, pp. 71–86, 2000.

[10] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao, "3D shapenets for 2.5 d object recognition and next-best-view prediction," *arXiv preprint arXiv:1406.5670*, vol. 2, no. 4, 2014.

[11] J. Daudelin and M. Campbell, "An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 3, pp. 1540–1547, 2017.

[12] H. Van Hoof, O. Kroemer, H. B. Amor, and J. Peters, "Maximally informative interaction learning for scene exploration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5152–5158.

[13] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, "Mobile bin picking with an anthropomorphic service robot," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 2327–2334.

[14] E. Arruda, J. Wyatt, and M. Kopicki, "Active vision for dexterous grasping of novel objects," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2881–2888.

[15] S. Chen, Y. F. Li, W. Wang, and J. Zhang, *Active sensor planning for multiview vision tasks*. Springer, 2008, vol. 1.

[16] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys (CSUR)*, vol. 35, no. 1, pp. 64–96, 2003.

[17] R. Pito, "A sensor-based solution to the "next best view" problem," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, vol. 1. IEEE, 1996, pp. 941–945.

[18] ——, "A solution to the next best view problem for automated surface acquisition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 10, pp. 1016–1030, 1999.

[19] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148–164, 2014.

[20] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1462–1468.

[21] R. Zeng, W. Zhao, and Y.-J. Liu, "Pc-nbv: A point cloud based deep network for efficient next best view planning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7050–7057.

[22] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[23] X. Han, Z. Zhang, D. Du, M. Yang, J. Yu, P. Pan, X. Yang, L. Liu, Z. Xiong, and S. Cui, "Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 234–243.

[24] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[26] F. Chazal and B. Michel, "An introduction to topological data analysis: fundamental and practical aspects for data scientists," *arXiv preprint arXiv:1710.04019*, 2017.

[27] H. Edelsbrunner and J. Harer, "Persistent homology-a survey," *Contemporary Mathematics*, vol. 453, pp. 257–282, 2008.

[28] https://github.com/robotic-vision-lab/Next-Best-View-Via-Topological-Features.

[29] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[30] W. J. Beksi and N. Papanikolopoulos, "A topology-based descriptor for 3D point cloud modeling: Theory and experiments," *Image and Vision Computing*, vol. 88, pp. 84–95, 2019.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[32] Open Source Robotics Foundation, *Gazebo Simulator*, 2021, http://gazebosim.org.

[33] ——, *Robot Operating System*, 2021, https://www.ros.org.