Multi-Sensor Next-Best-View Planning as Matroid-Constrained Submodular Maximization

Mikko Lauri[1], Joni Pajarinen[2], Jan Peters[3], and Simone Frintrop[1]

***Abstract*—3D scene models are useful in robotics for tasks such as path planning, object manipulation, and structural inspection. We consider the problem of creating a 3D model using depth images captured by a team of multiple robots. Each robot selects a viewpoint and captures a depth image from it, and the images are fused to update the scene model. The process is repeated until a scene model of desired quality is obtained. Next-best-view planning uses the current scene model to select the next viewpoints. The objective is to select viewpoints so that the images captured using them improve the quality of the scene model the most. In this paper, we address next-best-view planning for multiple depth cameras. We propose a utility function that scores sets of viewpoints and avoids overlap between multiple sensors. We show that multi-sensor next-best-view planning with this utility function is an instance of submodular maximization under a matroid constraint. This allows the planning problem to be solved by a polynomial-time greedy algorithm that yields a solution within a constant factor from the optimal. We evaluate the performance of our planning algorithm in simulated experiments with up to 8 sensors, and in real-world experiments using two robot arms equipped with depth cameras.**

***Index Terms*—Reactive and Sensor-Based Planning, RGB-D Perception, Multi-Robot Systems**

I. Introduction

SCENE reconstruction is the process of creating a digital model of a real-world scene from a set of images or other measurements of the scene. Models obtained via scene reconstruction are useful for robotic applications such as object manipulation, structural inspection [1], and waste sorting (Fig. 1). In an online setting, the model reconstructed from the images captured so far is used to plan from which viewpoint the next image is captured. Next-best-view (NBV) planning [2] determines the next viewpoint that provides the greatest improvement to the quality of the current model, reducing the amount of time and number of images required to reconstruct a model of desired quality.

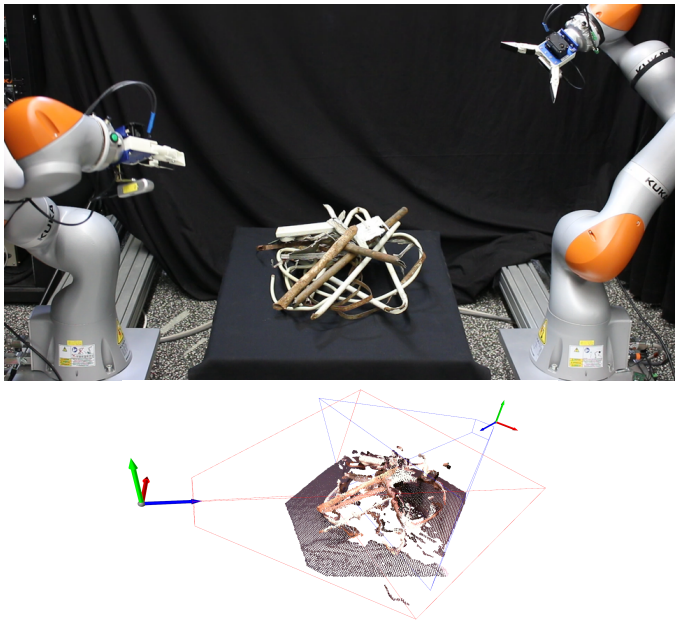


Fig. 1. Above: two robots sorting waste. Below: The robots create a 3D model by recording images from views around the scene (red and blue frustums). We propose an efficient method to find the next-best-views for multiple cameras based on greedy maximization of a submodular utility function. The views found by our algorithm avoid overlap between the sensors' fields of view.

Many tasks may benefit from deployment of a multi-robot team [3], however most approaches to NBV planning focus on the single-robot setting [2], [4], [5], [6], [7], [8]. In a multi-robot setting these approaches would plan the viewpoint for each robot individually and ignore coordination between team members. Time and resources are wasted if the same part of the scene is observed by multiple robots.

In this paper, we propose an efficient method for multi-sensor NBV planning that coordinates view selection and avoids overlapping views (Fig. 1). The constraint that each robot can choose at most one view is formalized as a *matroid*, a mathematical object that generalizes the concept of independence from linear algebra to sets. View selection and other related sensor selection problems often satisfy a diminishing returns property known as *submodularity* [9]: the benefit of adding a new view decreases with an increasing number of existing views. An advantage of submodular maximization under a matroid constraint is that a polynomial-time greedy algorithm provides a solution within a constant factor from the optimum [10].

Manuscript received: February 24, 2020; Revised May 22, 2020; Accepted June 19, 2020.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by European Research Council Grant No. 640554 (SKILLS4ROBOTS) and German Research Foundation project PA 3179/1-1 (ROBOLEAP).

[1]Mikko Lauri and Simone Frintrop are with Department of Informatics, University of Hamburg, Germany {lauri,frintrop}@informatik.uni-hamburg.de

[2]Joni Pajarinen is with Intelligent Autonomous Systems lab, TU Darmstadt, Germany and with Tampere University, Finland pajarinen@ias.tu-darmstadt.de

[3]Jan Peters is with Intelligent Autonomous Systems lab, TU Darmstadt, Germany and with Max Planck Institute for Intelligent Systems, Germany peters@ias.tu-darmstadt.de

Digital Object Identifier (DOI): see top of this page.

Unlike [11] who reconstruct regions of interest when communication is not constant, we aim to produce a complete scene reconstruction when robots communicate to coordinate their actions. Our approach is best suited for robots deployed near each other (Fig. 1) with reliable communication. Cui et al. [12] consider a single robot equipped with many sensors, one of which is chosen to be active at any time. In contrast, we consider joint planning where all sensors are operated simultaneously and the combination of views that is most useful is selected. We apply matroid-constrained submodular maximization similar to the multi-robot planning methods in [13], [14], [15]. We are the first to address multi-sensor NBV planning using submodularity.

Our key technical contribution is a utility function that avoids overlap between the views of multiple sensors. We prove that the utility function is monotonically increasing and submodular, and formulate multi-sensor NBV planning as submodular maximization under a matroid constraint. We describe an efficient greedy algorithm for the planning problem with an approximation guarantee. When the sensors' potential views are disjoint, our formulation reduces to solving independent single-sensor NBV planning problems. We experimentally verify the effectiveness of our proposed approach in a set of simulated experiments with up to 8 sensors, and in a real-world experiment with two robot arms.

The rest of the paper is organized as follows. We review related work in Section II, and define some useful mathematical concepts in Section III. We formulate the multi-sensor NBV problem in Section IV, and contrast it to single-sensor NBV planning. In Section V we introduce our proposed utility function, and prove that it is submodular. Section VI proposes a greedy algorithm for multi-sensor NBV planning. In Sections VII and VIII we report results from simulated and real-world experiments. Section IX concludes the paper.

II. Related work

We review related work in two areas: NBV planning in single and multi-sensor settings, and multi-robot planning using matroid constraints and submodularity.

A. Next-best-view planning

We focus on NBV planning approaches with a volumetric scene representation. A volumetric scene representation consists of a finite set of grid cells, or voxels. For each voxel, an occupancy probability that gives the probability of the voxel containing an obstacle is maintained. In NBV planning, the set of visible voxels from each candidate view is first estimated by applying raytracing. A score for the candidate view is calculated as the sum of per-voxel scores for each visible voxel. The NBV with the greatest score is selected.

NBV planning methods differ in how the score of a view is calculated. Counting measures [2], [4] count the total number of unknown visible voxels. Probabilistic measures employ quantities such as entropy of voxel occupancy, or the visibility probability of voxels. For example, [5] selects the view that has the greatest average occupancy entropy in the visible voxels. In [7], scores that weight per-voxel scores by the probability of the voxel being visible are proposed. Views that observe the most boundary voxels between known and unknown space are preferred in [16]. In [6], the method is extended to consider the uncertainty in sensor motion. Recently, [17] proposes to learn scoring of candidate views by using a 3D convolutional neural network with a multi-scale volumetric map representation. The assumption that voxel occupancies are independent is relaxed in [18] by applying Markov chain Monte Carlo.

Instead of a voxel grid, [8] proposes an implicit surface density representation. The representation consists of points observed on a surface. Regions with a high density of points are classified as core, and other regions as outliers. The expected observed volume between core and outlier regions is maximized. A Gaussian process (GP) implicit surface representation is used in [19]. The variance of the GP quantifies the uncertainty of the surface reconstruction. A planning algorithm uses the variance to find trajectories that reduce uncertainty the most.

The related problem of planning how a robot should manipulate an object within the view of a stationary camera is investigated in [20]. The object model is a signed distance function on a voxel grid. In [21], trajectories for environment exploration are scored by estimating the unknown volume visible along the trajectory. A robotic system for structural inspection is demonstrated in [1]. As a known environment is considered, an inspection path is planned offline to guarantee a desired amount of overlap between captured images.

Some recent works consider NBV planning in the multi-robot setting. Sukkar et al. [11] reconstruct regions of interest (ROIs) by controlling viewpoints of multiple robots equipped with cameras. ROIs correspond to fruit in an agricultural application and are detected by color thresholding. Candidate viewpoints are scored based on the expected information gain on the detected ROIs. The algorithm proposed in [3] is applied to decentralize the planning task so that each robot can plan its views without needing to constantly communicate with the other robots. Cui et al. [12] select if a robot should apply a laser range finder or depth camera next. Candidate sensing actions are scored by a weighted sum of the number of unknown voxels, occupied voxels, and voxels neighbouring a free voxel. Unlike [11], we target tasks where the robots communicate constantly to coordinate views. We do not focus on ROIs, but strive for a complete reconstruction. Different to [12], we choose the views of multiple sensors simultaneously. By applying submodular maximization, we provide a performance guarantee for our planning algorithm.

B. Matroid constraints and submodularity in planning

Multi-robot coordination may be viewed as an item selection task. Each robot selects an item (e.g., a trajectory), with the objective of maximizing a performance measure that is a function of the selected set of items. A matroid defines a system of independent sets, which models constraints such that one robot may select at most one trajectory. A submodular function defined on independent sets has a diminishing returns property that states that adding a new item to an existing set of items is less useful the larger the existing set is.

Maximizing a submodular function under a matroid constraint by a polynomial-time greedy algorithm provides a constant-factor approximation [10]. Furthermore, many information-theoretic functions are submodular, making greedy submodular maximization a popular approach for tasks such as planning sensor placements [9].

Multi-robot information gathering tasks such as exploration [14] and informative path planning [13] have been addressed as submodular maximization under a matroid constraint. A sequential greedy allocation (SGA) algorithm is proposed in [13], allowing extension of single-robot planning to any number of robots while maintaining performance guarantees. A distributed variant of SGA proposed in [14] scales up to larger problems due to distribution of the computation, while maintaining approximation guarantees. In [15], a greedy algorithm is proposed for coupled problems, such as selecting the composition of a multi-robot team and subsequently planning how the robots should act. The work most similar to ours is [14], where the distributed approach incurs an additional suboptimality penalty. In our paper we target the centralized setting where greedy submodular maximization enjoys a tighter suboptimality bound.

III. Preliminaries

We define now mathematical concepts that are later used.

A. Partition matroids

Matroids generalize the concept of independence from linear algebra to sets. For an introduction to matroids, we refer the reader to [22]. For the purposes of this paper, we only require the concept of a partition matroid.

Let A_i be n pairwise disjoint sets, and let a_i be integers s.t. $0 \leq a_i \leq |A_i|$. Denote by Ω the union of all A_i. Then $(\Omega, \mathcal{I})$ is a partition matroid if $\mathcal{I} = \{I \subseteq \Omega \mid \forall i : |I \cap A_i| \leq a_i\}$. The elements of $\mathcal{I}$ are *independent sets*. The sets A_i are *blocks* of the partition matroid. We assume $a_i = 1$ for all partition matroids, such that there is at most one element per block in an independent set.

B. Submodularity

Submodularity formalizes the notion of diminishing returns: the marginal utility of adding a new item to an existing set of items is smaller the larger the existing set is.

A set function $f : 2^{\Omega} \to \mathbb{R}$ is submodular if for any $A \subseteq B \subseteq \Omega$, and any $x \in \Omega \setminus B$, $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$. Additionally, f is monotonically increasing, if for any $A \subseteq \Omega$ and $x \in \Omega \setminus A$, $f(A \cup \{x\}) \geq f(A)$.

IV. Multi-sensor next-best-view planning

Consider a team of n sensors, and let $X_1, \ldots, X_n$ be n pairwise disjoint sets with X_i representing the possible views for sensor i. Denote by $\mathcal{I}$ the collection of independent sets in a partition matroid with blocks X_i. The problem of multi-sensor NBV planning is to select an independent set of views – one for each sensor – that maximizes the utility of the views.

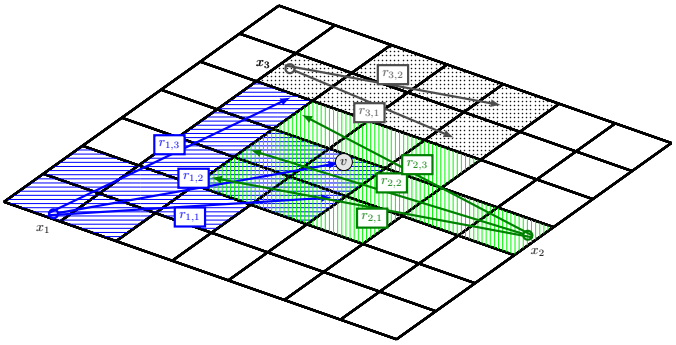


Fig. 2. Grid cells are shown by the black squares. Views are indicated by x_i. The rays $r_{i,j}$ emitted from x_i are indicated by colored arrows. The colored shading on a grid cell indicates that the grid cell is traversed through by a ray with the corresponding color.

Utility is measured by a function f mapping independent sets to a real number. Formally, the problem is

$$\max_{I \in \mathcal{I}} f(I). \tag{1}$$

Next, we introduce our environment and sensor models. We then discuss how utility is measured in single-sensor NBV planning, and motivate the need for a utility function f specifically designed for the multi-sensor setting.

A. Environment and sensor models

The environment is represented as a volumetric grid V with a finite number of grid cells. For each grid cell, $P(v)$ gives the probability that the grid cell is occupied by an obstacle. The cell occupancies are independent of each other.

Each of the sensors is a camera that outputs a depth image. The view x_i of each sensor is its translation and orientation w.r.t. a fixed coordinate system. A depth image is considered as a set of rays emitted from x_i. A ray is emitted in the direction of each pixel in the depth image, terminating after traversing a distance equal to the measured depth. We write each ray r as a sequence of grid cells it traverses through, e.g., $r = (v_1, v_2, \ldots, v_k)$, where $v_i \in V$ and k is the total number of grid cells traversed. At view x_i, the set of all rays emitted by the sensor is $R(x_i)$.

Fig. 2 illustrates a volumetric grid. For clarity the drawing is in two dimensions, but in this paper we deal with three dimensional grids. Three views and their corresponding sets of emitted rays are shown. For example, the set of rays emitted from x_1 is $R(x_1) = \{r_{1,1}, r_{1,2}, r_{1,3}\}$. Each ray $r_{i,j}$ is also a sequence of grid cells. For example, $r_{1,2}$ is the sequence of all grid cells traversed by the ray starting at the cell where x_1 is located and terminating at grid cell v labeled by the shaded circle marker near the center of the grid.

Given a real depth image recorded by a sensor, the occupancy probabilities are updated using the inverse sensor model of [23] as follows. Raytracing is applied to determine which grid cells are updated. The focal lengths (f_x, f_y) and the principal point (c_x, c_y) of the depth camera are required. Let d be the measured depth at pixel coordinates (x, y). The ray direction vector $\begin{bmatrix} (x-c_x)/f_x & (y-c_y)/f_y & 1 \end{bmatrix}^T$ is transformed to the coordinate system of the volumetric grid, and a ray of

length d is emitted in the resulting direction. The grid cells traversed by the ray are recorded. The grid cell where a ray terminates is observed as a hit, and other grid cells traversed by the ray are observed as misses. The log-odds representation for $P(v)$ is $L(v) = \log \frac{P(v)}{1-P(v)}$. Given an observed grid cell v and its observation z, the log-odds are updated to $L(v \mid z) = L(v) + l(z)$, where

$$l(z) = \begin{cases} \log \frac{p_h}{1-p_h} & \text{if } z = \text{hit} \\ \log \frac{p_m}{1-p_m} & \text{if } z = \text{miss} \end{cases} \tag{2}$$

and p_h and p_m are the hit and miss probabilities, respectively. The posterior $P(v \mid z)$ is obtained by inverting $L(v \mid z)$.

B. Score function for an emitted ray

The ray score function calculates the score of a single ray as a sum over the grid cells traversed by the ray. At each grid cell v, a weight term dependent on which other cells the ray traversed to reach v is multiplied by the information gain available at v. We formalize this by the following definition, and give concrete examples afterwards.

Definition 1 (Ray score function)**.** *Let r be a ray that traverses the sequence $(v_1, v_2, \ldots, v_k)$ of grid cells. The ray score function $s : V^k \to \mathbb{R}$ is defined as*

$$s(r) = \sum_{j=1}^{k} w_{j,r}(v_1, \ldots, v_{j-1}) c(v_j), \tag{3}$$

where $w_{j,r} : V^{j-1} \to \mathbb{R}$ is a weight term, and $c : V \to \mathbb{R}$ is the information gain available at a grid cell. For $j = 1$, we define $w_{1,r} \equiv 1$.

In our experiments we use the widely applied entropy score [5], [7] by setting the information gain equal to entropy of voxel occupancy, $c(v) = -P(v)\log_2 P(v) - (1 - P(v))\log_2(1 - P(v))$, and defining $w_{j,r}$ always equal to one. This choice encourages exploration by assigning a high score for views that observe voxels with a high uncertainty. However, we derive all our theoretical results for the general form given in the definition above, which includes many scores proposed in earlier literature. Counting measures [2], [4] are obtained by setting the weight term always equal to 1, and the information gain function to return 1 when the grid cell is unknown and 0 otherwise. Occlusion-aware scores [7] are obtained by setting the weight term equal to $\prod_{i=1}^{j-1}(1 - P(v_i))$, the probability that all grid cells traversed are free, i.e., the probability that the ray reaches v_{j-1} before hitting an obstacle and terminating. A region of interest $S \subset V$ is focused by setting the weight term to depend on the indicator function of S.

Single-sensor NBV approaches select a view x by maximizing $\sum_{r \in R(x)} s(r)$. A naive extension of the single-sensor NBV approach to a multi-sensor setting selects for each sensor i the view x_i that maximizes $\sum_{r \in R(x_i)} s(r)$. However, since there is no incentive for coordination between the sensors, unnecessarily overlapping views may be selected.

V. An Overlap-Aware Utility Function for Multi-Sensor Next-Best-View Planning

To coordinate view selection for multiple sensors, we propose an overlap-aware utility function to score sets of rays. For each grid cell, the utility function only takes into account the ray along which the weighted information gain is maximized. This captures the intuitive notion that overlap between multiple sensors should be avoided if possible, and only the most useful ray traversing through a particular grid cell should be considered in NBV planning. Our proposed utility function helps reduce unnecessary overlap in the selected views, as we later show experimentally. We prove our utility function is monotonically increasing and submodular, and show how single-sensor NBV planning arises as a special case when sensor views do not overlap.

A. Overlap-aware utility

Recall that X_i are the pairwise disjoint sets of available views for each sensor. Let Ω denote the set of all possible views obtained as the union of all X_i. The partition matroid of valid views is $(\Omega, \mathcal{I})$ with $\mathcal{I} = \{I \subseteq \Omega \mid \forall i : |I \cap X_i| \leq 1\}$. Any independent set $I \in \mathcal{I}$ contains at most one view from each X_i. The union of all rays emitted from views in an independent set is $R(I) = \cup_{x_i \in I} R(x_i)$. For example in Fig. 2, $R(\{x_1, x_2\}) = \{r_{1,1}, r_{1,2}, r_{1,3}, r_{2,1}, r_{2,2}, r_{2,3}\}$.

The overlap-aware utility function considers for each grid cell all the rays that traverse through the cell. For any grid cell v, we denote by $T_I(v)$ the subset of rays in $R(I)$ that traverse through v. Recalling that any ray $r \in R(I)$ is a sequence of grid cells, we define

$$T_I(v) = \{r \in R(I) | r \cap \{v\} \neq \emptyset\}. \tag{4}$$

For instance, letting $R(\{x_1, x_2\})$ be as in the paragraph above, and considering the grid cell v indicated in Fig. 2, $T_{\{x_1,x_2\}}(v) = \{r_{1,2}, r_{2,2}\}$, containing exactly the two rays that traverse through v.

We propose the following overlap-aware utility function.

Definition 2 (Overlap-aware utility function)**.** *Let $(\Omega, \mathcal{I})$ be a partition matroid of valid sensor views. For any independent set $I \in \mathcal{I}$ of views, the overlap-aware utility is*

$$f(I) = \sum_{v \in V} \max_{r \in T_I(v)} \left[w_{j,r}(v_1, \ldots, v_{j_r - 1}) c(v) \right], \tag{5}$$

where $w_{j,r}(v_1, \ldots, v_{j_r-1})$ is the weight term, and c is the information gain available at v. For $I = \emptyset$, set $f(\emptyset) = 0$.

At any grid cell, only the contribution from the ray that has the greatest weight term when reaching that grid cell is considered for overlap-aware utility. For instance, if the weight term is chosen as the probability that the ray reaches the grid cell, only the ray with the greatest probability contributes to the utility. Selecting overlapping views that observe the same grid cells is implicitly discouraged by the utility function. Emitting an additional ray that reaches a grid cell only improves the utility if the additional ray has a greater weight term. For example, consider the grid cell v indicated by the gray circle in Fig. 2. The sum term in Eq. (5) that corresponds to v is

the maximum over the two rays $r_{1,2}$ and $r_{2,2}$ that traverse through v.

B. Proof of submodularity and monotonicity

We prove a supporting lemma and then the main result.

Lemma 1. *For any $v \in V$, the set function*

$$g_v(I) = \max_{r \in T_I(v)} w_{j,r}(v_1, \ldots, v_{j_r-1}) c(v) \tag{6}$$

is monotonically increasing and submodular.

Proof. Fix $v \in V$ and $A \subseteq B \subseteq \Omega$. Thus, $T_A(v) \subseteq T_B(v)$, since adding more views can only increase the number of rays that traverse through v. The maximum cannot decrease, i.e., $g_v(A) \leq g_v(B)$, so g_v is monotonically increasing.

To prove submodularity, fix $v \in V$ and $A \subseteq B \subseteq \Omega$. Select an arbitrary view $x \in \Omega \setminus B$, which is not in B and thus also not in A. By the definition in Eq. (4), every element in $T_{\{x\}}(v)$ is in both $T_{A \cup \{x\}}(v)$ and $T_{B \cup \{x\}}(v)$. Thus,

$$T_{A\cup\{x\}}(v) = T_A(v) \cup T_{\{x\}}(v) \subseteq T_B(v) \cup T_{\{x\}}(v) = T_{B\cup\{x\}}(v),$$

which implies $g_v(A \cup \{x\}) = \max\{g_v(A), g_v(\{x\})\}$ and $g_v(B \cup \{x\}) = \max\{g_v(B), g_v(\{x\})\}$. Now,

$$\begin{aligned} &g_v(A \cup \{x\}) - g_v(A) = \max\{g_v(A), g_v(\{x\})\} - g_v(A) \\ &= \max\{0, g_v(\{x\}) - g_v(A)\} \geq \max\{0, g_v(\{x\}) - g_v(B)\} \\ &= \max\{g_v(B), g_v(\{x\})\} - g_v(B) = g_v(B \cup \{x\}) - g_v(B), \end{aligned}$$

where the inequality follows as g_v is monotonically increasing and the proof for submodularity is complete. □

Theorem 1. *The overlap-aware utility function f as defined in Eq. (5) is submodular and monotonically increasing.*

Proof. A sum of monotonically increasing submodular terms (Lemma 1) is monotonically increasing and submodular. □

Recall that a partition matroid with blocks X_i describes the valid combinations of views the robots may choose. We proved that the overlap-aware utility function f, Definition 2, is a submodular function of the independent sets of this partition matroid. In summary, multi-sensor NBV planning, Eq. (1) is matroid-constrained submodular maximization.

C. Single-sensor NBV planning is a special case

We prove that if there is no potential overlap between the views of any sensors, multi-sensor NBV planning reduces to solving n independent single-sensor NBV planning problems. Two views are *disjoint* if the rays emitted from the views do not traverse through any of the same grid cells.

Definition 3. *Two views x_i, x_j are disjoint if for every grid cell $v \in V$, the rays emitted from x_i and x_j do not overlap, that is, $\forall v : T_{\{x_i\}}(v) \cap T_{\{x_j\}}(v) = \emptyset$.*

In Fig. 2, the views x_1, x_3 are disjoint, and the views x_2, x_3 are disjoint, but the views x_1, x_2 are not disjoint.

The following proposition shows that if there is no overlap between the views of any two different sensors, the multi-sensor NBV problem with the overlap-aware utility function is equivalent to n single-sensor NBV planning problems.

Algorithm 1 Greedy multi-sensor NBV planning

Input: Partition matroid $(\Omega, \mathcal{I})$ of views
Output: Independent set $I_k \in \mathcal{I}$ containing k planned views

1: $k \leftarrow 0$, $I_k \leftarrow \emptyset$
2: **while** $\exists x \in \Omega : I_k \cup \{x\} \in \mathcal{I}$ **do**
3: $\quad x^* \leftarrow \operatorname{argmax}_{x \in \Omega \text{ s.t. } I_k \cup \{x\} \in \mathcal{I}} f(I_k \cup \{x\}) - f(I_k)$
4: $\quad I_{k+1} \leftarrow I_k \cup \{x^*\}$, $\Omega \leftarrow \Omega \setminus \{x^*\}$, $k \leftarrow k + 1$
5: **end while**
6: **return** I_k

Proposition 1. *Let X_i be pairwise disjoint sets of views for each sensor i. If for every $i \neq j$, all $x_i \in X_i$ and $x_j \in X_j$ are disjoint, then Eq. (1) with the overlap-aware utility function from Eq. (5) is equivalent to $\sum_{i=1}^{n} \max_{x_i \in X_i} f(\{x_i\})$.*

Proof. Let $I = \{x_1, \ldots, x_n\}$. As all views are disjoint, for any $v \in V$, there exists exactly one $x_i \in I$ such that $T_I(v) = T_{\{x_i\}}(v)$. The claim is proven by rearranging Eq. (5): $\sum_{i=1}^{n} \sum_{v \in V} \max_{r \in T_{\{x_i\}}(v)} [w_{j,r}(v_1, \ldots, v_{j_r-1}) c(v)] = \sum_{i=1}^{n} f(\{x_i\})$. □

VI. A GREEDY ALGORITHM FOR MULTI-SENSOR NBV PLANNING

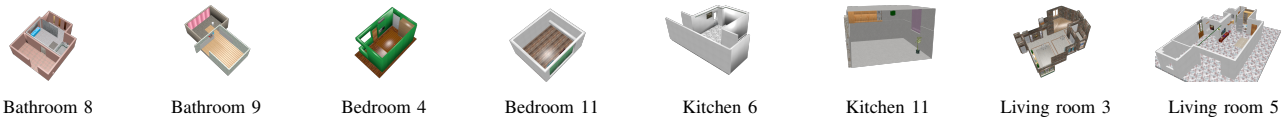
Algorithm 1 is a greedy strategy for solving Eq. (1). The algorithm starts from the initial solution $I_0 = \emptyset$ at iteration $k = 0$. As long as there exists a view x in the ground set Ω such that $I_k \cup \{x\}$ is an independent set, we repeat the following two steps. First, find the view x^* with the greatest marginal utility that maintains the matroid constraint, that is,

$$x^* = \operatorname*{argmax}_{x \in \Omega \text{ s.t. } I_k \cup \{x\} \in \mathcal{I}} f(I_k \cup \{x\}) - f(I_k). \tag{7}$$

Second, update the solution by $I_{k+1} = I_k \cup \{x^*\}$, remove x^* from the ground set, and increment k. When the input is the partition matroid of sensor views, the output I_n contains exactly one view for each of the n sensors. When f is monotonically increasing and submodular, I_n is a $\frac{1}{2}$-approximation [10], that is, $f(I_n) \geq \frac{1}{2} \max_{I \in \mathcal{I}} f(I)$.

We perform raytracing for each view $x \in \Omega$ once and store the best per-voxel scores $g_v(\{x\})$. We initialize $g_v(I_0)$ to zero for all voxels. The marginal utility of a view x on Line 3 is then computed as follows. Initialize the marginal utility to zero. Loop over voxels visible for x and compare the stored value $g_v(\{x\})$ to the old value $g_v(I_k)$. If the stored value is greater than the old value, the positive difference is added to the marginal utility. If the old value is greater than the stored value, continue to the next visible voxel. After the view x^* with the greatest marginal utility is recovered, the old values for visible voxels are updated to $g_v(I_{k+1}) = \max\{g_v(I_k), g_v(\{x^*\})\}$. If raytracing for one view has computational complexity $O(R)$, the overall computational complexity of Algorithm 1 is $O(R|\Omega| + n|\Omega|)$ where the second term accounts for computing the marginal utilities of up to $|\Omega|$ views for up to n sensors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.3007445, IEEE Robotics and Automation Letters



Bathroom 8 | Bathroom 9 | Bedroom 4 | Bedroom 11 | Kitchen 6 | Kitchen 11 | Living room 3 | Living room 5

Fig. 3. The types of layouts in the simulation experiments.

TABLE I
AVERAGE AREA UNDER CURVE FOR SURFACE COVERAGE IN THE SIMULATION EXPERIMENTS. BEST AVERAGE VALUES BOLDED.

	Method	Avg	Bathroom 8	Bathroom 9	Bedroom 4	Bedroom 11	Kitchen 6	Kitchen 11	Living room 3	Living room 5
2 cameras	Ours	**80.1**	82.3	79.0	76.1	85.1	82.6	77.4	78.7	77.0
	Single	75.2	77.1	73.7	72.0	80.3	77.5	72.7	73.0	71.3
	Random	72.0	71.2	70.4	72.1	75.8	74.2	75.6	70.6	69.4
4 cameras	Ours	**89.2**	90.5	88.6	86.7	92.1	90.7	87.6	87.5	86.7
	Single	84.5	85.9	83.6	82.6	87.9	86.1	82.7	82.2	81.4
	Random	83.9	82.2	81.9	83.8	86.0	84.7	86.3	81.8	81.1
8 cameras	Ours	**94.6**	95.1	94.2	93.4	95.9	95.4	94.3	93.1	92.9
	Single	91.9	92.5	91.3	91.0	93.6	92.8	91.4	89.8	89.4
	Random	91.4	90.4	90.0	91.6	92.5	91.9	93.2	89.6	89.3

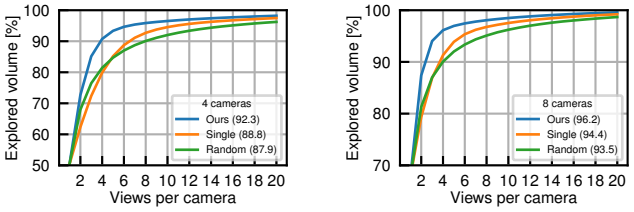


Fig. 4. Volume explored as a function of the number of views per camera for 4 cameras (left) and 8 cameras (right) in the simulation experiments. Note the different vertical axis scales.

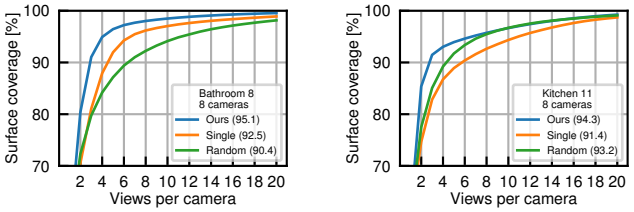


Fig. 5. Surface coverage as a function of the number of views per each of 8 cameras in layouts Bathroom 8 (left) and Kitchen 11 (right) in the simulation experiments.

The proposed greedy submodular maximization algorithm provides an approximation guarantee and runs in polynomial time. Eq. (1) could also be solved by exhaustive search or a heuristic algorithm. Exhaustive search recovers an optimal solution but has a worst-case computational complexity of $O(R|\Omega| + |\Omega|^n)$ as the utility of every independent set must be evaluated. Heuristic algorithms that start with an initial solution and modify it locally are also applicable. Examples of such algorithms include genetic optimization algorithms, tabu search, or simulated annealing.

VII. SIMULATION EXPERIMENTS

We evaluate our NBV planning algorithm in synthetic environments and in a robotic setup. This section presents the results for the synthetic environments.

A. Experimental setup

We use the validation set of SceneNet RGB-D [24], a collection of rendered RGB-D images from randomly generated synthetic environments. The validation set consists of 1000 scene configurations. Each scene configuration depicts a synthetic room layout with randomly sampled objects in physically plausible poses. The room layout is one of 8 possible choices[1] illustrated in Fig. 3. Each scene configuration has 300 labeled cameras poses along a continuous trajectory. 320-by-240 pixel depth images from the labeled poses are available to compute a ground truth scene reconstruction.

We consider $n = 2$, 4, or 8 sensors, and split the camera poses in each scene configuration into n disjoint subsets X_i. Each subset contains possible views for sensor i and spans the entire trajectory. We plan a sequence of 20 views for each sensor by Algorithm 1. We compare to independent single-sensor NBV planning of views as described in Subsection IV-B, and to selecting the next views uniformly at random. Each experiment is repeated 10 times. The true depth images are applied to update occupancy probabilities as described in Subsection IV-A. We use hit and miss probabilities $p_h = 0.9$ and $p_m = 0.1$, respectively. We use a voxel grid with a resolution of 0.05 m per voxel. The initial occupancy probability is $P(v) = 0.5$ for all voxels. To vary the prior information about the scene, the first views are sampled randomly in each experiment and are the same for all methods. We use a resolution of 0.1 rays per pixel and a maximum range of 10 m for raytracing.

We record the explored volume and the surface coverage as a function of the number of views per camera. We compare these values to the ground truth model created from all images. For computing the surface coverage, a point in the ground truth model is considered observed if there is a point closer than 0.05 m to it in the reconstruction.

[1]We omit the "Office 14" layout as it has only one scene configuration.

B. Results

Fig. 4 shows for 4 and 8 cameras the average fraction of explored volume as a function of number of views. The numbers in parentheses in the legend show the micro-averaged area under curve (AUC) value of the respective methods. Our algorithm reaches 90% explored volume with between 2-3 fewer views per camera compared to the other two methods. Single-sensor planning does not coordinate view selection of multiple sensors, and prefers strongly overlapping views in the first 2-4 steps. Random view selection has no preference for such views, and thus performs better during these steps. After strongly overlapping views are removed from future consideration, single-sensor planning avoids selecting useless views on subsequent steps and outperforms random over the entire span of 20 views. Our algorithm avoids overlapping views, outperforming single-sensor planning and random view selection for any number of selected views.

Table I shows the overall micro-averaged AUC for surface coverage, and the AUC for each layout. On average, our algorithm observes more surface points than single-sensor planning or randomly selecting views. As the number of cameras increases, the amount of potential overlap between the views increases. With 4 or 8 cameras, the performance of single-sensor planning and random are similar, while ours performs significantly better. This shows our algorithm is able to successfully coordinate views with increasing amount of potential overlap. Our method performs best in layouts with many rooms and occluding walls, such Bathroom 8, Kitchen 6, and Living room 5 (see Fig. 3). The improvement is smallest in layouts consisting of a single room such as Kitchen 11, especially when the number of cameras is 8. Single planning with 4 or 8 cameras also performs almost identically to random in both of the living room layouts.

Fig. 5 shows the surface coverage with 8 cameras in Bathroom 8 and Kitchen 11. Numbers in parentheses in the legend show the AUC. Our method performs best for any number of views in the Bathroom 8 layout that has many connected rooms separated by walls that occlude views (see Fig. 3). In the Kitchen 11 layout consisting of a single room with no occluding walls, our method still performs best with less than 8 views per camera, and then equally well as random view selection. Single-sensor planning selects strongly overlapping views and performs worse than randomly selecting views.

For both planning algorithms, raytracing to compute the per-voxel scores takes most of the runtime. The average runtime was 0.8 s per candidate view.

VIII. Real-world experiments

We set up three scenes: a scene with light clutter (Fig. 6, left), the same scene with a large obstacle added (Fig. 6, right), and a scene with unsorted metal waste (Fig. 1, top). Intel RealSense D435 depth cameras are attached to the two KUKA LBR iiwa robot arms. For each robot, we sample 20 candidate views around the workspace. In each view, the camera points to the center of the workspace. We plan a sequence of 8 views by the proposed multi-sensor NBV planning (Ours), single-sensor NBV planning (Single), or random view selection (Random). As there are only two cameras, for Ours we find the independent set that maximizes overlap-aware utility directly by evaluating all combinations. Other experimental settings are as in Section VII.

TABLE II
Average unknown volume (cm^3) and its 95% confidence interval in the real-world scenes. Bolded values show all statistically significantly best-performing methods.

Scene	Method	After 3 views	After 5 views	After 8 views
Clutter	Ours	**11404 ± 128**	**7016 ± 55**	**5064 ± 55**
	Single	**11592 ± 89**	**7104 ± 222**	**4888 ± 55**
	Random	18844 ± 3267	10018 ± 1009	5869 ± 426
Obstacle	Ours	**17360 ± 400**	**13592 ± 55**	**11628 ± 83**
	Single	**17696 ± 277**	**13556 ± 6**	**11492 ± 128**
	Random	24627 ± 3960	15954 ± 1226	**12064 ± 390**
Waste	Ours	**14296 ± 133**	**6400 ± 0**	**4476 ± 39**
	Single	16584 ± 33	9260 ± 139	4984 ± 11
	Random	20274 ± 2887	10549 ± 1144	5210 ± 547

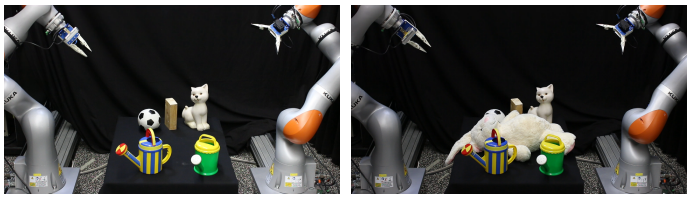


Fig. 6. The clutter scene (left) and the obstacle scene (right).

Table II shows the average unknown volume[2] (lower is better) and its 95% confidence interval, after 3, 5, and 8 views in each scene. In the "Waste" scene, Ours outperforms Single. The views in the scene overlap strongly, and our method avoids redundant views. In the "Clutter" and "Obstacle" scenes, there is no significant difference between Ours and Single. In the "Obstacle" scene, the difference of Single, Ours, and Random after 8 views is not significant. Randomly selecting 8 views is sufficient for a good reconstruction. In all cases, the random baseline performs worst.

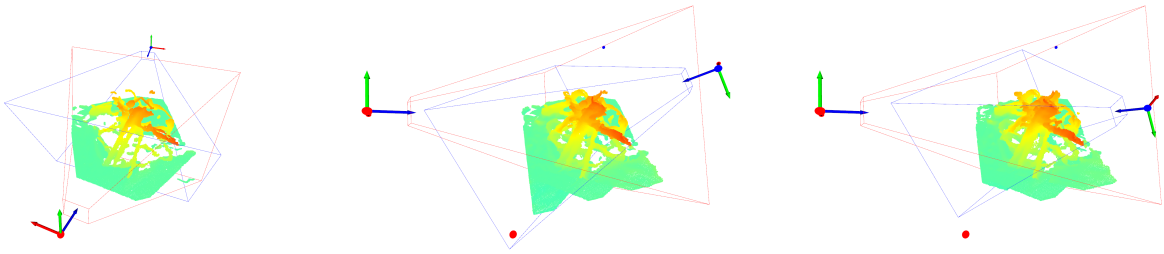
In all scenes, Ours selected the same sequence of views in each repetition. This was the case for Single as well. In Table II the variation for Ours and Single is due to unpredictable events such as localization or sensor noise.

Fig. 7 qualitatively compares Ours and Single. Our method explicitly considers that the two sensors' fields of view overlap, and selects views with less overlap. Single maximizes utility for each sensor independently, and selects views that overlap resulting in less explored volume. After the two views, the remaining unknown volume was 34 000 cm^3 (Ours) and 38 000 cm^3 (Single).

IX. Discussion and Conclusion

Several topics beyond the scope of this paper remain unexplored. Future work could investigate how the granularity of the candidate view sampling affects performance. Considering more candidate viewpoints is guaranteed to improve

[2]Unknown volume never reaches zero as object insides are unobservable.



(a) Initial view (b) Ours (c) Single

Fig. 7. Qualitative results in the "Waste" scene. The left subfigure shows the initial reconstruction and views (red and blue frustum). The middle and right subfigures show the second views selected by Ours and Single, respectively. The reconstruction color corresponds to height above the tabletop. The coordinate axes indicate the current pose of each sensor. Compared to Single, our method selects a view for sensor 2 (blue frustum) that avoids overlap with sensor 1 (red frustum) and observes the region to the bottom left.

performance, but it might not be beneficial in practice due to the increased computational cost. Dynamic generation of candidate viewpoints could also be considered, e.g., similar to [21]. Many formulations of information gain have been proposed and compared in single-sensor object reconstruction [7], while we applied only the entropy score. Comparing information gain formulations in the multi-sensor case is a direction for future work. The demands for NBV planning vary depending on the application [8], [11], [12], and studying the quality of the reconstructions with respect to the parameters of the algorithm can help fit our proposed method to specific applications.

In conclusion, we propose a monotonically increasing and submodular overlap-aware utility function for multi-sensor next-best-view planning with a volumetric environment representation. Our greedy planning algorithm is guaranteed to produce a solution within a constant factor from the optimum. Experimental results show the algorithm avoids planning overlapping views, outperforms independent single-sensor planning and random view selection methods, and can be implemented in a real-world robotic system.

REFERENCES

[1] J. Quenzel, M. Nieuwenhuisen, D. Droeschel, M. Beul, S. Houben, and S. Behnke, "Autonomous MAV-based indoor chimney inspection with 3D laser localization and textured surface reconstruction," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1-2, pp. 317–335, 2019.

[2] C. Connolly, "The determination of next best views," in *ICRA*, 1985, pp. 432–435.

[3] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

[4] 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.

[5] 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.

[6] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View/state planning for three-dimensional object reconstruction under uncertainty," *Autonomous Robots*, vol. 41, no. 1, pp. 89–109, 2017.

[7] 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.

[8] R. Border, J. D. Gammell, and P. Newman, "Surface Edge Explorer (SEE): Planning Next Best Views Directly from 3D Observations," in *ICRA*, 2018, pp. 6116–6123.

[9] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[10] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions – II," in *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.

[11] F. Sukkar, G. Best, C. Yoo, and R. Fitch, "Multi-Robot Region-of-Interest Reconstruction with Dec-MCTS," in *ICRA*, 2019, pp. 9101–9107.

[12] J. Cui, J. T. Wen, and J. Trinkle, "A Multi-Sensor Next-Best-View Framework for Geometric Model-Based Robotics Applications," in *ICRA*, 2019, pp. 8769–8775.

[13] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[14] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Autonomous Robots*, vol. 43, no. 2, pp. 485–501, 2019.

[15] J. Liu and R. K. Williams, "Submodular optimization for coupled task allocation and intermittent deployment problems," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3169–3176, 2019.

[16] 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.

[17] B. Hepp, D. Dey, S. N. Sinha, A. Kapoor, N. Joshi, and O. Hilliges, "Learn-to-score: Efficient 3D scene exploration by predicting view utility," in *ECCV*, 2018, pp. 437–452.

[18] L. Hou, X. Chen, K. Lan, R. Rasmussen, and J. Roberts, "Volumetric Next Best View by 3D Occupancy Mapping using Markov Chain Gibbs Sampler for Precise Manufacturing," *IEEE Access*, vol. 7, pp. 121 949–121 960, 2019.

[19] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.

[20] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *ICRA*, 2011, pp. 5031–5037.

[21] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration," in *ICRA*, 2016, pp. 1462–1468.

[22] J. Oxley, "What is a matroid?" *CUBO, A Mathematical Journal*, vol. 5, no. 3, pp. 176–215, 2003.

[23] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[24] J. McCormac, A. Handa, S. Leutenegger, and A. J.Davison, "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?" in *ICCV*, 2017, pp. 2678–2687.