An Information Gain Formulation for Active Volumetric 3D Reconstruction

Stefan Isler, Reza Sabzevari, Jeffrey Delmerico and Davide Scaramuzza

Abstract— **We consider the problem of next-best view selection for volumetric reconstruction of an object by a mobile robot equipped with a camera. Based on a probabilistic volumetric map that is built in real time, the robot can quantify the expected information gain from a set of discrete candidate views. We propose and evaluate several formulations to quantify this information gain for the volumetric reconstruction task, including visibility likelihood and the likelihood of seeing new parts of the object. These metrics are combined with the cost of robot movement in utility functions. The next best view is selected by optimizing these functions, aiming to maximize the likelihood of discovering new parts of the object. We evaluate the functions with simulated and real world experiments within a modular software system that is adaptable to other robotic platforms and reconstruction problems. We release our implementation open source.**

SUPPLEMENTARY MATERIAL

The accompanying video and software package are available at: `http://rpg.ifi.uzh.ch`.

I. INTRODUCTION

Object reconstruction in three dimensions is an important step in robust perception and manipulation tasks. This work considers the problem of reconstructing an object that is unknown a priori, but is spatially bounded. We assume that we obtain dense 3D input data from a camera-based sensor, but do not restrict to a particular modality (i.e. stereo, monocular structure-from-motion, structured light, etc.). A mobile robot positions the sensor for different views of the volume containing the object, with the goal of generating a complete volumetric model in as little time as possible. We take an *active vision* approach, and select each *next best view* (NBV) using feedback from the current partial reconstruction. In particular, we consider the information that we expect to gain from a new viewpoint in choosing an optimal NBV from a set of candidates. To encourage the progress of the reconstruction, we define this information to be higher for views that contribute more to the reconstruction. In other words, a view with more information is one where more of the previously unobserved surface becomes visible.

Since we consider an unknown object, the information that can be gained from a new view is not known a priori and must be estimated online at the time of the reconstruction. To do so, the robot must reason about observed and unobserved areas of the object as well as about possible occlusions. The use of a volumetric model facilitates such visibility considerations. There, the visibility of a surface from a given camera position can be inferred by casting rays into the voxel space. To account for sensor uncertainties, modern volumetric models are probabilistic. We define the *information gain* (IG) in a probabilistic volumetric map as the sum of expected information enclosed in smaller volumes (voxels), that are likely to be visible from a particular view. The information contained in voxels is denoted as *volumetric information* (VI). Previous works on this problem [1], [2] typically combine the IG with additional, often system-specific terms designed to optimize the process with regard to constraints such as movement cost or data registration and quality. In a system-agnostic reconstruction problem, it is the information gain formulation and the choice of possible view point candidates that remain as the two most important factors to the reconstruction. The possible view point candidates are highly dependent on the robotic system, since the general 6 DoF search space is constrained by its kinematics. Information gain, on the other hand, can be used in arbitrary robotic systems that maintain a probabilistic volumetric map. This paper focuses on evaluating possible formulations for information gain based on variations of the volumetric information.

We approach the autonomous reconstruction task as an iterative process consisting of the three largely independent parts (i) 3D model building, (ii) view planning, and (iii) the camera positioning mechanism, as observed in [3]. The orthogonality of the involved tasks has inspired us to design our own autonomous reconstruction system in a modular manner to allow for fast reconfigurations of the software for different applications and robotic setups. We utilize the Robot Operating System (ROS) [4] software framework, which allows a hardware-agnostic design through the use of its interprocess communication interfaces. Within this framework, we use off-the-shelf components for the 3D model building and camera positioning sub-tasks, and focus only on view planning based on our proposed IG formulations.

A. Related Work

Research on the *Next-Best-View problem* and conceptually similar problems in *Active Vision* dates back several decades [5], [6]. The most frequently referenced surveys of the field include an overview of early approaches by Scott et al. [7] and an overview of more recent work by Chen et al. [8]. We will follow the categorization introduced by

All the authors are with the Robotics and Perception Group, University of Zurich, Switzerland (`http://rpg.ifi.uzh.ch`). Reza Sabzevari is currently with Robert Bosch GmbH, Corporate Research. This research was funded by the the Swiss National Science Foundation through the National Center of Competence in Research Robotics (NCCR) and the UZH Forschungskredit.

Scott et al. in distinguishing between model-based and non-model-based reconstruction methods.

Model-based methods assume at least an approximate a priori model of the scene, e.g. from aerial imagery [9]. They rely on knowledge of the geometry and appearance of the object, which may not be available in many real world scenarios.

Non-model based approaches use relaxed assumptions about the structure of the object, but the required information for planning the next best view must be estimated online based on the gathered data. The method used to reason about possible next actions depends on the environment representation in which the sensor data is registered. Scott et al. distinguished between surface-based and volumetric approaches, and more recently methods have been proposed that employ both [1]. In a surface-based approach, new view positions are evaluated by examining the boundaries of the estimated surface, e.g. represented by a triangular mesh [10], [11]. This approach can be advantageous if the surface representation is also the output of the algorithm because it permits examination of the quality of the model while constructing it. The disadvantage is that visibility operations on surface models are more complicated than with volumetric ones.

Volumetric approaches have become popular because they facilitate visibility operations and also allow probabilistic occupancy estimation. Here, the gathered data is registered within a volumetric map consisting of small volumes, called voxels, that are marked either with an occupancy state or an occupancy probability. View positions are evaluated by casting rays into the model from the view position and examining the traversed voxels, therefore simulating the image sampling process of a camera. The information gain metric that quantifies the expected information for a given view is defined on this set of traversed voxels.

One method to create an IG metric is to count traversed voxels of a certain type. Connolly et al. [12] and Banta et al. [13] count the number of unknown voxels. Yamauchi et al. [14] introduced the concept of *frontier voxels*, usually defined as voxels bordering free and unknown space, and counted those. This approach has found heavy use in the exploration community where the exploration of an unknown environment is the goal, rather than reconstruction of a single object [15].

The research of Vasquez-Gomez et al. [2] is a recent example where a set of frontier voxels is used for reconstruction. They count what they call *occplane voxels* (short for occlusion plane), defined as voxels bordering free and occluded space. Another method is to employ the entropy concept from information theory to estimate expected information, as shown in [1]. This necessitates the use of occupancy probabilities but has the advantage that the sensor uncertainty is considered. Potthast et al. [16] argue that the likelihood that unknown voxels will be observed decreases as more unknown voxels are traversed and that this should be considered in the information gain calculation. They model the observability using a Hidden Markov Model and introduce empirically found state transition laws to calculate posterior probabilities in a Bayesian way.

B. Contributions

In this paper, we propose a set of information gain formulations (IG) and evaluate them along with recent formulations in the literature. Our IG formulations are obtained by integrating novel formulations for volumetric information (VI):

- **Occlusion Aware VI**: Quantifies the expected visible uncertainty by weighting the entropy within each voxel by its visibility likelihood.
- **Unobserved Voxel VI**: Restricts the set of voxels that contribute their VI to voxels that have not been observed yet.
- **Rear Side Voxel VI**: Counts the number of voxels expected to be visible on the back side of already observed surfaces.
- **Rear Side Entropy VI**: Quantifies the expected amount of VI as defined for the Occlusion Aware VI, but restricted to areas on the rear side of already observed surfaces.
- **Proximity Count VI**: This VI is higher the closer an unobserved voxel lies to already observed surfaces.

We evaluate all of these VIs in real and synthetic experiments, based on following criteria: The amount of discovered object surface (surface coverage), the reduction of uncertainty within the map, and the cost of robot motion.

Finally, we release our modular software framework for active dense reconstruction to the public. The ROS-based, generic system architecture enables any position controlled robot equipped with a depth sensor to carry out autonomous reconstructions.

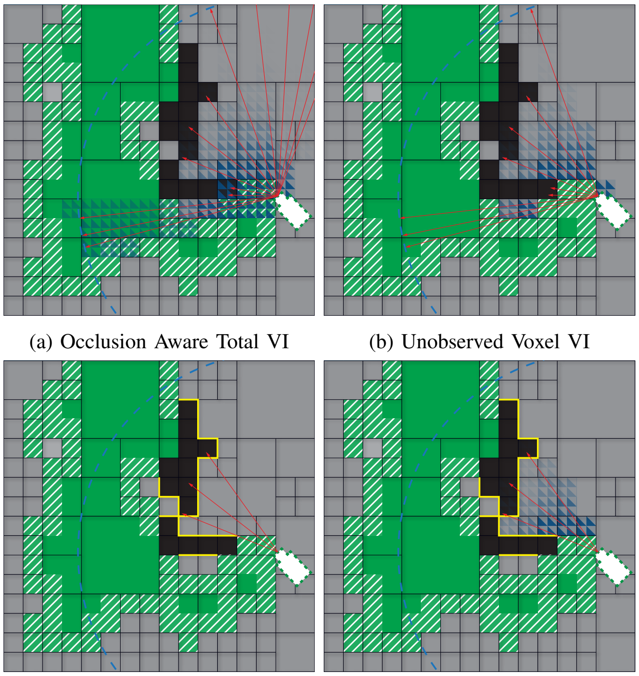
C. Paper Overview

After introducing our proposed volumetric information formulations in Section II, we give a short insight into our generic system architecture in Section III. Experiments and results are shown in Section IV. In Section V we summarize and discuss our results.

II. Volumetric Information

We define the *Volumetric Information* (VI) as a formulation for information enclosed in a voxel. In this context, the *Information Gain* (IG) is the amount of information (i.e. VI) that each view candidate is expected to provide for the 3D reconstruction. IG will be used as a metric to find the *Next Best View* (NBV) from a set of candidate sensor positions, which in object-centric reconstruction tasks are usually sampled from simple geometries like the cylinder in [10] or the sphere in [17]. The NBV with respect to the reconstruction is the view that provides the most gain in surface coverage and uncertainty reduction.

Let $\mathcal{V}$ be the set of sensor positions. A set of points is sampled from the projection of the 3D object on view $v \in$


(a) Occlusion Aware Total VI

(b) Unobserved Voxel VI

(c) Rear Side Voxel VI

(d) Rear Side Entropy VI

Fig. 1. Visualization of the IG function with different VI formulations in 2D on an exemplary state of the map: The map shows occupied (black), unknown (grey) and empty (green) regions and a view candidate (white camera). Additionally frontier voxels (striped white), unknown object sides (yellow), considered ray sets (red), maximal ray length (dashed blue circle) and VI weights (opacity of blue triangles) are shown.

$\mathcal{V}$. Let $\mathcal{R}_v$ be the set of rays cast through the sampled points on view v. Representing the 3D world as a volumetric cube-based grid of uniform size, each ray traverses through a set of voxels $\mathcal{X}$ before it hits the object's surface. We denote the predicted IG for view $v \in \mathcal{V}$ as $\mathcal{G}_v$. The cumulative volumetric information $\mathcal{I}$, collected along all rays, depends on the VI formulation that is used:

$$\mathcal{G}_v = \sum_{\forall r \in \mathcal{R}_v} \sum_{\forall x \in \mathcal{X}} \mathcal{I} \tag{1}$$

Figure 1 visualizes four of the proposed formulations for an exemplary 2D scenario. The images show a snapshot of a possible state in the map during reconstruction and how IG is estimated: Each voxel has an assigned occupancy likelihood estimated based on registered point measurements. Depending on the likelihood, a voxel's state is considered to be occupied (black), free (green) or unknown (gray). Other states like proximity to the frontier and occplane voxels or the rear side of surfaces can be identified during the ray casting operation. Using the state as input we quantify VI and then integrate it to compute a measure of IG. The following sections present a set of VI formulations using indicator functions, probabilistic functions or a combination of both.

A. Occlusion Aware

The volumetric information within a voxel can be defined as its entropy:

$$\mathcal{I}_o(x) = -P_o(x)\ \ln P_o(x) - \overline{P}_o(x)\ \ln \overline{P}_o(x) \tag{2}$$

where $P_o(x)$ is the probability of voxel x being occupied, and $\overline{P}_o$ denotes the complement probability of P_o, i.e. $\overline{P}_o = 1 - P_o$.

We further consider the visibility likelihood P_v of a voxel and write:

$$\mathcal{I}_v(x) = P_v(x)\, \mathcal{I}_o(x) \tag{3}$$

For a voxel x_n we have:

$$P_v(x_n) = \prod_{i=1}^{n-1} \overline{P}_o(x_i) \tag{4}$$

where $x_i, i = 0 \ldots n-1$ are all the voxels traversed along a ray before it hits voxel x_n.

Plugging Eq. 3 into Eq. 1, the IG given for a particular view $v \in \mathcal{V}$ estimates the entropy within the visible volume of the map. Refer to Fig. 1a for a visualization. Using this volumetric information, the next best view is the one with the highest visible uncertainty. Using Eq. 4, a voxel further away from the sensor position contributes less to the IG, accounting for its higher likelihood of occlusion by unobserved obstacles. One flaw of such a definition for information is that occupancy and unoccupancy of a voxel are equally weighted, neglecting that free voxels are not contributing to the reconstruction task.

B. Unobserved Voxel

Voxel state is commonly defined as a binary variable [2], [18]. We can set up an indicator function based on the observation state of the voxels (known or unknown), such that:

$$\mathcal{I}_u(x) = \begin{cases} 1 & x \text{ is unknown} \\ 0 & x \text{ is known} \end{cases} \tag{5}$$

Including the occupancy likelihood $\mathcal{I}_v(x)$, as in Eq. 3, into Eq. 5 that defines the voxel state we have:

$$\mathcal{I}_k(x) = \mathcal{I}_u(x)\, \mathcal{I}_v(x) \tag{6}$$

$\mathcal{I}_k(x)$ measures the hidden information in unobserved voxels. This is visualized in Fig. 1b.

C. Rear Side Voxel

The volumetric information formulated in Eq. 6 does not consider if an unobserved voxel is likely occupied or not. Assuming that the object of interest is bigger than a single voxel, the voxels traced along a ray at a certain distance directly behind an occupied voxel are likely to be occupied. A ray incident on the rear side of an already observed surface frontier is certain to be incident on an unobserved part of the object.

Let set $\mathcal{S}_o$ be the set of *rear side voxels*, defined as occluded, unknown voxels adjacent on the ray to an occupied voxel, thus we have:

$$\mathcal{I}_b(x) = \begin{cases} 1 & x \in \mathcal{S}_o \\ 0 & x \notin \mathcal{S}_o \end{cases} \tag{7}$$

Fig. 1c visualizes this type of VI.

D. Rear Side Entropy

Unlike the rear side voxel count in Eq. 7, we can consider the occupancy likelihood for all the unknown voxels behind an occupied voxel. Thus, $\mathcal{S}_o$ in Eq. 7 will be the set of all unknown voxels behind an occupied voxel instead of the set of rear side voxels, so we have:

$$\mathcal{I}_n(x) = \mathcal{I}_b(x)\,\mathcal{I}_v(x) \tag{8}$$

This type of VI is shown in Fig. 1d.

Equations 7 and 8 are computationally efficient because they allow a search for views that are likely pointed at unobserved parts of the object without additional evaluations, but only by examining the impact point of rays. The disadvantage is that these formulations only consider a ray's direction towards, and not the proximity to, already observed surface frontiers of the object. A sensor position pointing at the object from the side might not feature any ray that hits a rear side, even though its rays traverse the previously occluded volume behind observed surfaces in close proximity to the observed surface and are therefore likely to hit a surface voxel as well.

E. Proximity Count

To formulate a VI that considers the proximity of occluded voxels to the surface, we augment our volumetric map: when new measurements are integrated into the model, we continue following the rays behind intersected surface voxels until a distance d_{max}, and mark each of these occluded voxels with their distance $d(x)$ to the surface voxel. If a voxel is already marked, we keep the smaller distance. We use this distance information to define the volumetric information as follows:

$$\mathcal{I}_p(x) = \begin{cases} d_{max} - d(x) & x \text{ is unknown} \\ 0 & x \text{ is known} \end{cases} \tag{9}$$

The VI is higher if many voxels close to already observed surfaces are expected to be observed. A disadvantage is that this may lead to high rated sensor positions pointing away from the object through regions previously occluded that failed to be registered as empty space during later data registrations.

F. Combined

In an attempt to combine properties of different VI we explore a VI formulation based on the set $\mathcal{H}$ of those defined in the previous subsections, i.e. $\mathcal{I}_{combined} = f(\mathcal{I}^{h_0}, \mathcal{I}^{h_1}, \dots)$, where $I^h \in \mathcal{H}$. Since the characteristics of the proposed IGs are varied, there is no derivable formula to consolidate the different formulations. We evaluate a weighted linear combination, such that:

$$\mathcal{I}_c = \sum_{h \in \mathcal{H}} w_h\, \mathcal{I}^h \tag{10}$$

where w_h is the weight corresponding to $\mathcal{I}^h$. Such weights could be learned offline and loaded through object recognition from a database using e.g. a visual vocabulary, but that is beyond the scope of this work.

III. A Generic System Architecture for Active Dense Reconstruction

We evaluate our VI formulations within a flexible software system that is adaptable to different robotic platforms through its modular design. Its structure is an extension of the abstraction proposed in [3]. We divide the *modeling part* into three subtasks, reduce *view planning* to the NBV decision only, and divide the *camera positioning mechanism* into two layers, yielding the following independent components to take part in the reconstruction, as depicted in Figure 2:

- The **Sensor** module is responsible for data acquisition.
- The **Reconstruction Module** carries out vision algorithms for 3D sensing, e.g. stereo-matching or monocular depth estimation.
- The **Model Representation** includes *data integration* and *IG calculation*.
- The **View Planner** acts as a control module, collecting data about the reconstruction process and taking the *NBV decision*.
- The **Robot Interface Layer** defines the interface between View Planner and robot specific code, providing *view feasibility* and *cost calculations*.
- The **Robot Driver** implements the Robot Interface and includes all code specific to the hardware platform.

The view planner controls the iterative reconstruction process, consisting of a cycle of the following steps: *data retrieval*, *data processing*, *data integration*, *viewpoint generation and evaluation (NBV planning)* and *acting*. It communicates exclusively with the *Robot Interface* and the *Model Representation*. The exchanged data includes the proposed and evaluated views along with the IG values $\mathcal{G}_v$ and robot movement costs $\mathcal{C}_v$, which are provided by the robot and intended to constrain its movement. For NBV evaluation we combine them in the utility function $\mathcal{U}_v$:

$$\mathcal{U}_v = \frac{\mathcal{G}_v}{\sum_{\mathcal{V}} \mathcal{G}} - \frac{\mathcal{C}_v}{\sum_{\mathcal{V}} \mathcal{C}} \tag{11}$$

with $\sum_{\mathcal{V}} \mathcal{G}$ and $\sum_{\mathcal{V}} \mathcal{C}$ the total IG and cost respectively predicted to be obtainable at the current reconstruction step over all view candidates. The NBV v^* is found by maximizing $\mathcal{U}$ over all views v:

$$v^* = \arg\max_{v} \mathcal{U}_v \tag{12}$$

The reconstruction ends when the highest expected information gain of a view falls below a user-defined threshold g_{thresh}, i.e. if

$$\mathcal{G}_v < g_{thresh} \quad \forall v \in \mathcal{V} \tag{13}$$

View candidates can be provided either by the Robot Interface (RI) or the Model Representation (MR). Both techniques hold advantages: The Robot Interface can use the robot's kinematics to build a set of feasible sensor positions, avoiding the expensive inverse kinematics calculations necessary if the view candidates are generated

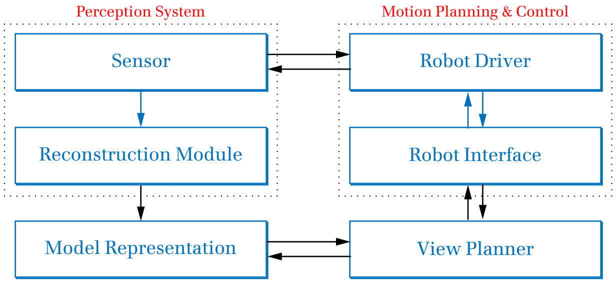


Fig. 2. Framework Overview: Main modules and their communication interfaces (arrows) are visualized.

by the Model module. The Model representation holds all knowledge of the world and can use it to create a set of target-oriented sensor positions. Only the sensor and robot interfaces need to be implemented for use with a new robot platform.

In summary, the View Planner iterates through the following steps:

1) Collect data from the sensors.
2) Request a view candidate set from the RI Layer or MR.
3) Optionally prefilter the view candidate set to evaluate IG and cost only on a subset.
4) Request cost for each view candidate from RI Layer.
5) Request IG for each view candidate from MR.
6) Calculate the utility function combining IGs and costs.
7) Determine NBV.
8) Check if termination criterion is fulfilled.
9) If criterion is not fulfilled: Command RI Layer to move the sensor to the NBV, then repeat the procedure.

IV. Experiments

Information Gain based on VI is a metric used as an indicator to estimate which next view will be most informative to the reconstruction. An informative view maximizes (i) the amount of new object surface discovered and (ii) the uncertainty reduction in the map. Additionally, (iii), we are interested in constraining the robotic motion to facilitate data registration using overlap in the obtained data and to save energy. We therefore evaluate our VI formulations on these three criteria.

A. Simulation

The reconstruction scene for the simulation consists of an object placed on a textured ground within four walls (Fig. 3). Around the object we generate a set of 48 candidate views, distributed uniformly across a cylinder with a half-sphere on top, such that they face the model from different poses. We use three models that we texturized, all of which are available online: the Stanford bunny and dragon[1] as well as a teapot, as visible in Fig. 4. The robot is a free-flying stereo camera with 6 DoF, with which we can carry out unconstrained movements. For a simulation environment, we use *Gazebo*[2], for which stereo processing can be carried out using ROS[3]. All reconstructions start by placing the stereo camera at a defined initial position, facing the object. Computed pointclouds are integrated into a probabilistic volumetric map based on OctoMap [19]. The map has a resolution of 1 cm and 0.8 is used as the lower likelihood bound for occupied, 0.2 as the upper likelihood bound for empty voxels. Views are removed from the candidate set once visited to keep the algorithm from getting stuck in positions for which the IG is overestimated. We do not use the stopping criterion but instead run 20 iterations for each trial.

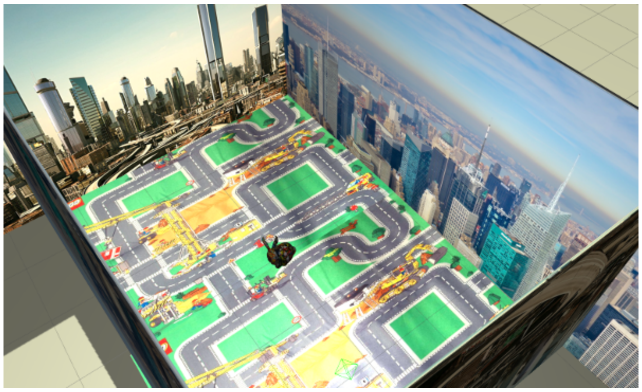
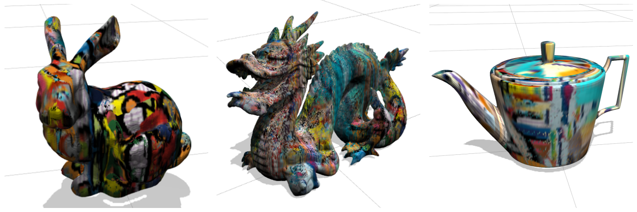


Fig. 3. Simulation reconstruction scene



(a) Stanford bunny (b) Stanford dragon (c) Teapot

Fig. 4. Synthetic model datasets

To quantify the reconstruction progress in terms of surface coverage, we compare the pointcloud models obtained during reconstruction with the pointcloud of the original model. For each point in the original model the closest point in the reconstruction is sought. If this point is closer than a registration distance[4] the surface point of the original model is considered to have been observed. The surface coverage c_s is then the percentage of observed surface points compared to the total number of surface points of the model:

$$\text{Surface coverage } c_s = \frac{\text{Observed surface points}}{\text{Surface points in original model}} \quad (14)$$

To calculate the total entropy we consider a bounding cube with 1.28 m side length around the object and define the total entropy to be

$$\text{Entropy in map} = \sum \text{Entropy of voxels within cube} \quad (15)$$

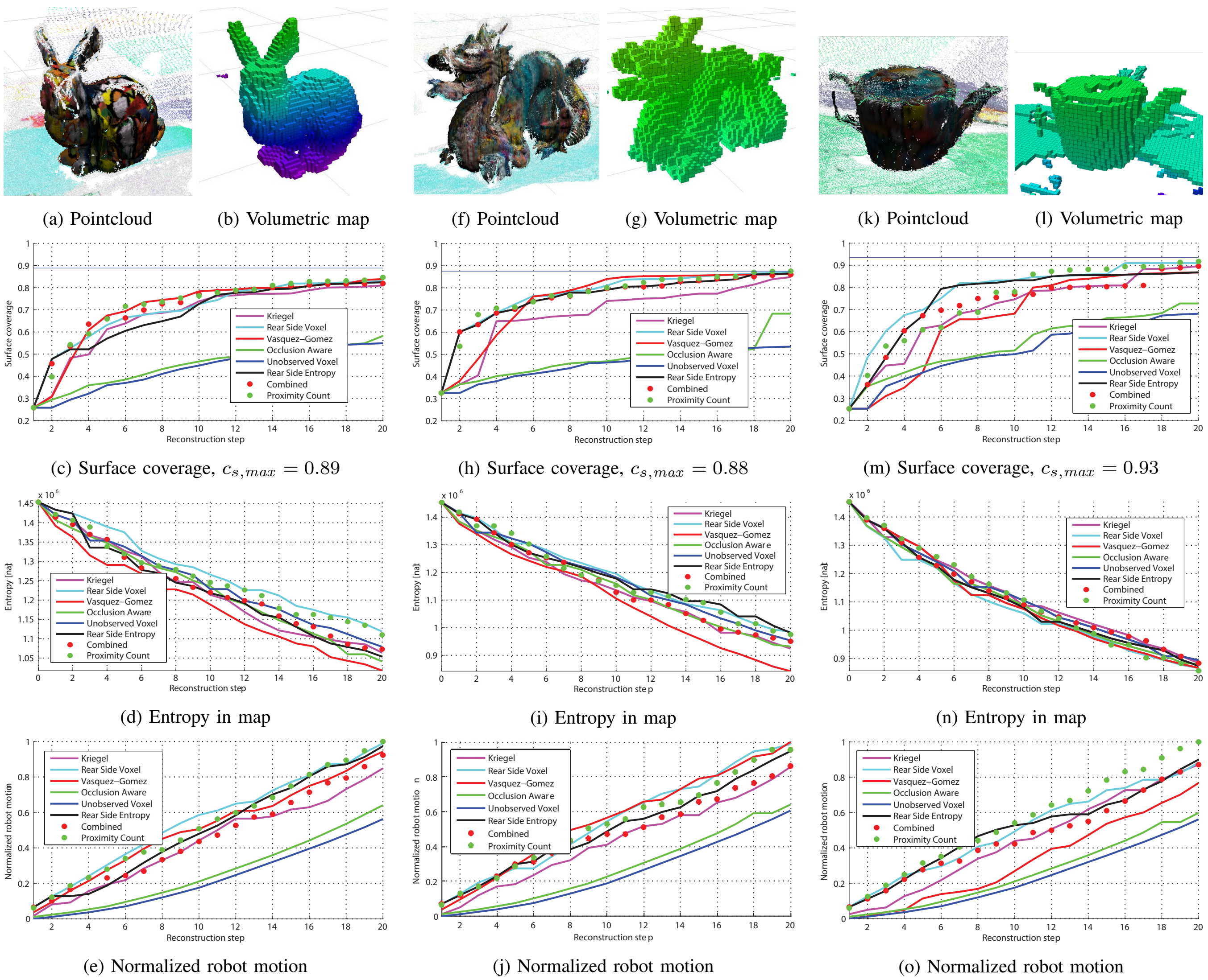
The robot motion is defined as the Euclidean distance between sensor positions, normalized with the maximal

[1] Available from the Stanford University Computer Graphics Lab.

[2] http://www.gazebosim.org

[3] We use the *stereo_img_proc* package.

[4] We chose $d_{reg} = 8mm$.



(a) Pointcloud (b) Volumetric map (f) Pointcloud (g) Volumetric map (k) Pointcloud (l) Volumetric map

(c) Surface coverage, $c_{s,max} = 0.89$ (h) Surface coverage, $c_{s,max} = 0.88$ (m) Surface coverage, $c_{s,max} = 0.93$

(d) Entropy in map (i) Entropy in map (n) Entropy in map

(e) Normalized robot motion (j) Normalized robot motion (o) Normalized robot motion

Fig. 5. Reconstructions in simulation: Evaluation of the reconstruction results for the Stanford bunny (a-e), Stanford dragon (f-j) and the teapot (k-o) datasets. We compare our methods to the methods of Kriegel [1] and Vasquez-Gomez [2].

distance moved by the robot in our experiments:

$$\text{Norm. robot motion} = \frac{\sum \text{Euclidean distances moved}}{\text{Maximal total distance}} \quad (16)$$

We compare our formulations to the IG methods of Kriegel [1] and Vasquez-Gomez [2]. Kriegel neglects possible occlusions and directly integrates Eq. 2 in Eq. 1 to obtain the total entropy, which they average over the total number of traversed voxels n:

$$\mathcal{G}_{v,Kriegel}(v) = \frac{1}{n} \sum_{\forall r \in \mathcal{R}_v} \sum_{\forall x \in \mathcal{X}} \mathcal{I}_o(x) \quad (17)$$

Vasquez-Gomez defines desired percentages $\alpha_{des,oc} = 0.2$ and $\alpha_{des,op} = 0.8$ of occupied and occplane voxels in the view, respectively, and bases his IG formulation on how close the expected percentages α_{oc} and α_{op} are:

$$\mathcal{G}_{v,Vasquez}(v) = f(\alpha_{oc}, \alpha_{des,oc}) + f(\alpha_{op}, \alpha_{des,op}) \quad (18)$$

with

$$f(\alpha, \alpha_{des}) = \begin{cases} h_1(\alpha, \alpha_{des}) & \text{if } \alpha \leq \alpha_{des} \\ h_2(\alpha, \alpha_{des}) & \text{if } \alpha > \alpha_{des} \end{cases} \quad (19)$$

where

$$h_1(\alpha, \alpha_{des}) = -\frac{2}{\alpha_{des}^3}\alpha^3 + \frac{3}{\alpha_{des}^2}\alpha^2 \quad (20)$$

and

$$\begin{aligned} h_2(\alpha, \alpha_{des}) = &- \frac{2}{(\alpha_{des}-1)^3}\alpha^3 + \frac{3(\alpha_{des}+1)}{(\alpha_{des}-1)^3}\alpha^2 \\ &- \frac{6\alpha_{des}}{(\alpha_{des}-1)^3}\alpha + \frac{3\alpha_{des}-1}{(\alpha_{des}-1)^3} \end{aligned} \quad (21)$$

$f(\cdot)$ is equal to one if the estimated percentage matches the desired percentage.

We present the results in Fig. 5. Plots (5c), (5h) and (5m) show the surface coverage achieved when using the different metrics on the three models. The surface coverage

c_s is calculated on the complete model surface, but not all parts of a model's surface are actually observable. This is in part due to occlusions, for instance on the bottom side where it touches the ground. But this can also be due to the block matching algorithm[5] failing to estimate the depth for all areas, e.g. because of lacking texture or suboptimal lighting conditions. The maximally achievable surface coverage is shown as a horizontal blue line in our plots, calculated by having a camera visiting all view candidates.

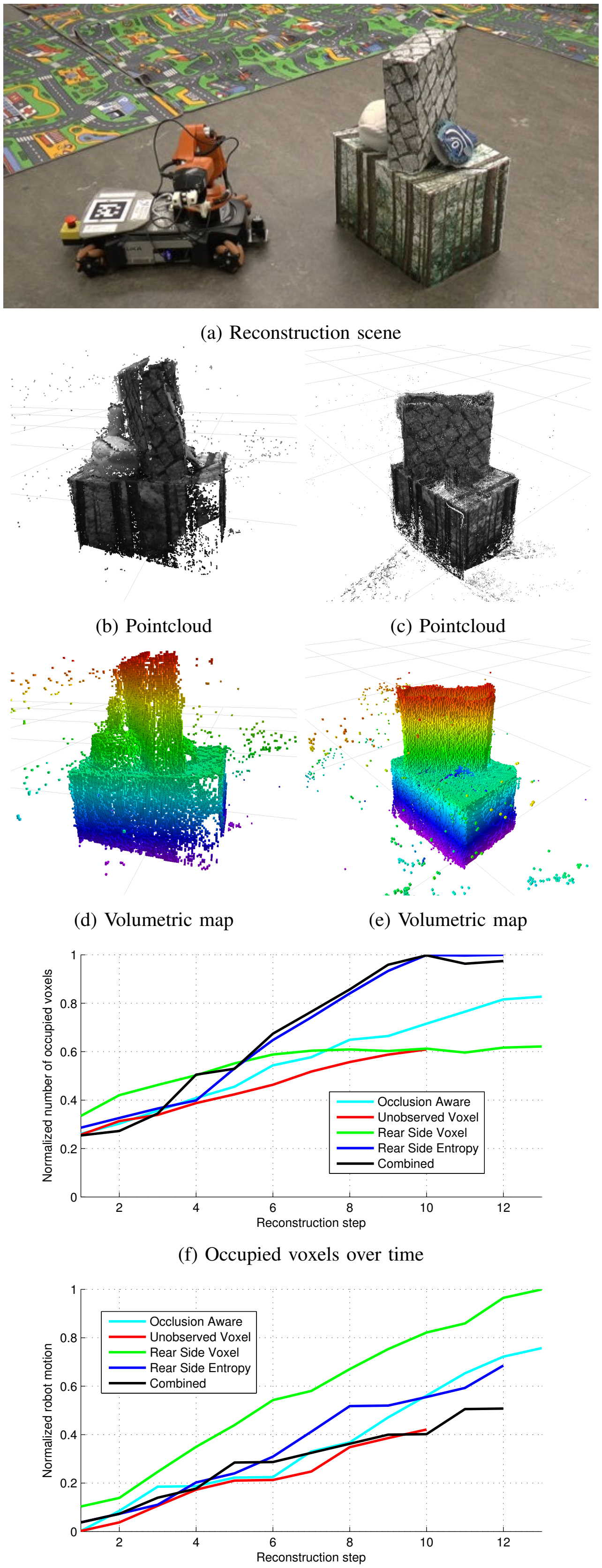
When comparing the performance of the different formulations, the Rear Side Voxel, Proximity Count and Combined VI appear to have a slight advantage in surface coverage speed if compared over all three experiments. In the presented results, the Combined VI consists of Kriegel's VI with $w_{Kriegel} = 30$ and the Rear Side Entropy VI with $w_n = 1$. The choice of participating VIs in this combination is designed to fuse an entropy-based formulation with a proximity-based one. The weights were found empirically, however they were set in order to approximately balance terms with different average magnitudes. While this linear combination of VIs performs very well, outperforming its components at times, the determination of the weighting coefficients is an open problem. Offline learning methods on large object sets might be a suitable solution, requiring the addition of an object recognition module and database.

Considering the performance with respect to entropy reduction, Vasquez-Gomez and perhaps Kriegel appear to have an advantage over our methods, though entropy reduction as shown in plots (5d), (5i) and (5n) is less discriminative than surface coverage; It takes place steadily at comparable rates for all formulations. Note that entropy reduction also results from observing unknown but free space in the map, where no new information about the object is obtained. More object-oriented VI formulations such as ours discover more new object surface at the expense of freeing less space, possibly resulting in less entropy reduction.

Robot motion cost, as shown in plots (5e), (5j) and (5o), is not more discriminative than entropy reduction. Combined VI and Kriegel's method perform best by a small margin if we neglect Occlusion Aware and Unobserved Voxel VI. The latter two appear to be too constrained in motion, yielding low cost but causing a degradation of their reconstruction performance.

B. Real World

We show the setup of our real world experiments in Fig. 6a. We equip a KUKA Youbot[6], a mobile robot with an omnidirectional base and a 5 DoF arm, with a global shutter RGB camera[7] on its end-effector. The vision pipeline consists of SVO [20], a vision-based, monocular SLAM algorithm, which estimates the camera motion, and REMODE [21], an algorithm for real-time, probabilistic



(a) Reconstruction scene

(b) Pointcloud

(c) Pointcloud

(d) Volumetric map

(e) Volumetric map

(f) Occupied voxels over time

(g) Normalized robot motion

Fig. 6. Real world reconstruction scene and results

[5] Refer to opencv.org for specifics about the algorithm.

[6] http://www.youbot-store.com

[7] Matrix Vision Bluefox mvBlueFox-IGC200wc.

monocular dense reconstruction, which is able to estimate depth for both strongly and weakly textured surfaces. To retrieve depth information from a given view, the robot moves the camera around the view that serves as key frame until REMODE reports successful convergence of depth estimates. While the vision pipeline and robot driver run on the on-board NVIDIA Jetson TK1, the map, IG calculations and NBV decisions are carried out on an Intel i7 desktop machine, but online in real time.

We present reconstruction results for two different scenes in Fig. 6. To evaluate the performance of different VI formulations, we cannot use the surface coverage as there is no ground truth data available. Instead, we report the normalized number of occupied voxels, defined as

$$\text{Norm. nr. of occupied voxels} = \frac{\text{Occupied voxels}}{\text{Maximal nr. of occupied voxels}} \tag{22}$$

Results are shown in Fig. 6. Comparisons must be carried out with caution because exact repetition of initial start positions cannot be guaranteed for different runs. In the real experiments the Rear Side Entropy and Combined VI outperformed all of our other formulations by a significant margin. They also performed well with respect to movement cost, which we define in these experiments as a combination of the movements of the base and the arm. For the Combined VI as defined by Eq. 10, we used the following weights: $w_o = 1$, $w_u = 20$, $w_b = 10$ and $w_n = 10$. As in the simulated trials, these values were chosen empirically to balance VI formulations of different magnitudes.

V. Conclusion

We have proposed and evaluated novel VI formulations that can be used for next best view decisions in volumetric reconstruction tasks. They are shown to yield successful reconstructions efficiently. Our results also show that including the visibility likelihood of voxels when estimating the entropy from view candidates does not improve performance for object-centric reconstruction tasks if no other means to focus VI on the object are employed. When considering particularly cluttered reconstruction scenes, visibility considerations might offer performance benefits, but this is left for future work.

Considering the likelihood of rays cast from view candidates to hit part of the object on the other hand yields very good results. We propose to find rays with a high likelihood by evaluating if we expect it to hit the backside of an already observed surface or to quantify the likelihood by considering the proximity of traversed unobserved voxels to already observed surfaces. This yields more object-focused VI formulations and less observations of the space surrounding the reconstruction target. We argue that the task of finding obstacle free space around the object for robot movement should be separated from the reconstruction itself. It is therefore desirable to have the most object-focused VI formulations possible.

The ROS-based, generic active dense reconstruction system used in this work is made available for public use.

References

[1] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects," *J. of Real-Time Image Processing*, pp. 1–21, 2013.

[2] J. Vasquez-Gomez, L. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next best view planning for 3d object reconstruction with positioning error," *Int. J. of Advanced Robotic Systems*, vol. 11, p. 159, 2014.

[3] L. Torabi and K. Gupta, "An autonomous six-DOF eye-in-hand system for in situ 3D object modeling," *Int. J. of Robotics Research*, vol. 31, no. 1, pp. 82–100, 2012.

[4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 2, 2009, p. 5.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 333–356, 1988.

[6] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[7] W. Scott, G. Roth, and J.-F. Rivest, "View planning for automated 3d object reconstruction inspection," *ACM Computing Surveys*, vol. 35, no. 1, 2003.

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

[9] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixa, M. Suppa, and G. Hirzinger, "View planning for multi-view stereo 3D reconstruction using an autonomous multicopter," *J. of Intelligent and Robotic Systems*, vol. 65, no. 1-4, pp. 309–323, 2012.

[10] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 10, pp. 1016–1030, 1999.

[11] S. Chen and Y. Li, "Vision sensor planning for 3-D model acquisition," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 5, pp. 894–904, 2005.

[12] C. Connolly *et al.*, "The determination of next best views," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2. IEEE, 1985, pp. 432–435.

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

[14] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 1997, pp. 146–151.

[15] J. Wettach and K. Berns, "Dynamic frontier based exploration with a mobile indoor robot," in *Int. Symp. on Robotics (ISR)*. VDE, 2010, pp. 1–8.

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

[17] M. Trummer, C. Munkelt, and J. Denzler, "Online next-best-view planning for accuracy optimization using an extended e-criterion," in *Int. Conf. on Pattern Recognition (ICPR)*. IEEE, 2010, pp. 1642–1645.

[18] P. S. Blaer and P. K. Allen, "Data acquisition and view planning for 3-d modeling tasks," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 417–422.

[19] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010.

[20] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 15–22. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2014.6906584

[21] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 2609–2616. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2014.6907233