J. Vis. Commun. Image R. xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A probabilistic framework for next best view estimation in a cluttered environment

Christian Potthast *, Gaurav S. Sukhatme

Robotic Embedded Systems Laboratory, University of Southern California, Los Angeles, CA 90089-0781, United States

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Next best view estimation
Sensor placement
Sensor planning
View planning
Robot exploration
3-D perception
Cluttered environments
Missing points

ABSTRACT

In this article, we present an information gain-based variant of the next best view problem for occluded environment. Our proposed method utilizes a belief model of the unobserved space to estimate the expected information gain of each possible viewpoint. More precise, this belief model allows a more precise estimation of the visibility of occluded space and with that a more accurate prediction of the potential information gain of new viewing positions. We present experimental evaluation on a robotic platform for active data acquisition, however due to the generality of our approach it also applies to a wide variety of 3D reconstruction problems. With the evaluation done in simulation and on a real robotic platform, exploring and acquiring data from different environments we demonstrate the generality and usefulness of our approach for next best view estimation and autonomous data acquisition.

1. Introduction

Major research areas in computer vision and robotics are autonomous 3D data acquisition and reconstruction of dense geometry of full environments or single objects. A substantial difficulty, and of most importance, is the selection of viewpoints for data acquisition and to accomplish subsequent tasks like reconstruction or object manipulation. Oftentimes we are limited by time, energy, or storage, but also occlusion making it necessary to estimate appropriate viewpoints for data acquisition. Over the past decade many algorithms have been developed to make this process autonomous and efficient for 2D [1] as well as 3D [2] environments. A widely used technique for automatic data acquisition and exploration is to iteratively compute and place the sensor to a new observation position where a new sensing action is performed. From a pure reconstruction point of view, we face the problem of optimizing view selection from the set of available views. The data, from which we want to reconstruct, is usually presented as a video or as individual images, showing the world or object in question from different viewpoints [3,4]. From this data, we want to find the subset of data that can best and most efficiently reconstruct the environment. In general, the estimation of such viewpoints known as the *Next Best View* (NBV) problem, first discussed in [5], which seeks a single additional sensor placement or viewpoint given a heuristic or reward function.

In many cases sensing actions are time consuming, but also an increase in data does not guarantee a better reconstruction result, one seeks a sequence of viewpoints which observe the environment in the minimum number of actions or images. How many measurements are needed to observe the environment depends, in principle on the selected viewpoints and with that the gain in knowledge about the world. The potential knowledge increase of new observation positions can be estimated by reasoning about current world information. In practice however, making precise estimations of the knowledge increase is difficult and error prone, because the estimation involves predicting the unobserved portions of the world.

This is especially true in cluttered environments, because they impose many visibility constraints due to occlusion thereby increasing the difficulty of making accurate predictions of the knowledge gain. Inaccurate estimations lead to over or under prediction of the gained information, hence possibly non-optimal observation positions are chosen. The more accurate we predict the information gain the fewer observation are necessary to fully observe the environment.

In this paper, we describe our work on automatic data acquisition and view point selection by iteratively estimating the knowledge gain of potential new observation positions. Our approach to estimate knowledge gain in cluttered environments is to reason about the unobserved space using probabilistic methods. Intuitively, we try to model the fact that the deeper we look into unob-

* Corresponding author.
E-mail addresses: potthast@usc.edu (C. Potthast), gaurav@usc.edu (G.S. Sukhatme).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.07.006

served regions, the higher the probability of observing a hidden object residing in it. Our method chooses the next best viewing position that promises the highest expected knowledge gain.

To evaluate our approach we use a robotic system to actively move the sensor to new viewpoints. However, our approach is not restricted to a robotic system and can be used for a variety of different tasks including data subset selection for reconstruction. We compare our approach to different robot exploration strategies, which iteratively estimate the NBV position and position the sensor to the new position.

The reminder of the paper is organized as follows. In Section 2 we examine related research. Next, we formalize our approach in Section 3. The details of our system architecture and point cloud preprocessing are described in Section 4. Finally, we show experimental results in Section 5 followed by a discussion of the results in Section 6.

2. Related research

View planning approaches are used in a variety of different research ares. There are two major surveys of the problem, one describing the sensor planning problem in vision applications [6]. The other is a more recent survey of view planning for 3D object reconstruction and inspection [7]. Of particular relevance to our work are approaches for exploration and automatic data acquisition using a robotic system. A survey for robot environment mapping and exploration can be found in [8]. In this brief review of related work, we focus on different approaches to estimate a set of potential new viewing positions as well as estimating the best position out of this set.

The main differences between existing approaches are the selection of the potential new viewing positions and the estimation of the NBV of this set. The earliest NBV estimation [5] proposed two approaches to create a complete model of a scene. The new sensor position is chosen either by simulating viewpoints sampled from the surface of a sphere and scored by using ray-casting and change estimation in unknown space or by using a histogram of normals for unexplored octree node faces.

Several methods have been developed which generate the potential viewpoints uniformly randomly in the workspace. In [9] an exploration algorithm was developed for a robot operating in 2D but acquired environment information in 3D. Potential new viewing positions are randomly generated on the 2D workspace of the robot and evaluated for their individual information gain. The potential new viewing positions are sampled uniformly randomly in a safe region [1], which consists of free space bounded by solid curves. To estimate the NBV position of this sample set a cost function trading off information gain and travel cost is evaluated and applied to each sample point. Similarly, in [10] the potential viewing positions are sampled randomly from a list of line segments bordering unobserved space. However, compared to [1] the utility function which evaluates the viewing position is more complex and takes the precision of localization into account. This ensures better localization and registration of the individual measurement. In [2] a probabilistic surface reconstruction approach is described using a depth cameras and a robotic manipulator holding an object. New viewing positions are uniformly sampled around the object and the manipulator hand. Given a current surface reconstruction a metric is formulated to score the potential new scanning positions in terms of reduction of uncertainty from virtual created depth maps.

Various methods are developed for reconstruction of dense geometry. In [4] the next best view estimation is used for efficient 3D model creation from images. The system estimates new viewpoints that best reduces the reconstructed model uncertainty. Haner and Heyden [3] and Hornung et al. [11] proposes a method for 3D reconstruction from a set of unordered images. The goal is to estimate a subset of images by sequentially estimating new viewpoint and select the best match. Their approach reduces the computation time and increases accuracy of the reconstructed model. To find and select new viewpoints the authors in [12,13] utilize surface normal information. Specifically, Pito [12] introduces the notion of positional space, which allowed, in combination with a detailed scanner model, refraining of expensive ray-casting methods for the next best view estimation. In [13], Massios and Fisher developed a quality constraint which estimated the quality of seen voxels in the scenes by utilizing the surface normal information. This improves the overall reconstruction of the measurement since viewpoints are selected which improve the laser range data quality of the scanned surface. The work of Haner and Heyden [14] proposes a continuous optimization of method of finding many future next best view positions to reconstruct the captured environment. The system utilizes a real-time visual simultaneous localization and mapping (SLAM) approach to estimate the sensor position and map the environment.

Camera in hand system used for model building are used by Trummer et al. [15] and Wenhardt et al. [16]. Both methods rely on images and feature tracking for measurement registration and model reconstruction. In [15] the authors present a combined online approach of tracking, reconstruction and NBV estimation. In their work the extended e-criterion is used, which guides the sensor to the next position to minimize the reconstruction uncertainty. Feature tracking and 3D reconstruction in [16] is done using a Kalman filter. The optimal NBV, which improves the reconstruction accuracy, is estimated by reducing the uncertainty in the state estimation.

A variety of frontier-based approaches have been developed that generate potential new viewing positions on the boundary between free and unobserved space. The earliest [17] uses a probabilistic occupancy grid map to represent and reason about the environment. Adjacent frontier edge cells are grouped together into frontier regions. The centroids of these groups form the set of potential new viewing positions. The closest frontier region position is the NBV position. Frontier regions are also used in [18] to build a detailed 3D outdoor map in a two step process. First a coarse exploration of the environment is performed by utilizing a 2D map and observability polygons to compute coverage positions. In the second step a 3D view planner samples potential view positions which lie on the frontier region. For each of the candidate viewpoints they count the number of visible cells using a ray-traversal algorithm. In the work of [19] the authors build a complete framework and estimate the NBV position given localization quality, navigation cost and information gain. The information gain is computed over the entropy of the region around potential new NBV positions. Stachniss and Burgard [20] introduces coverage maps to better represent the environment. They propose different methods to estimate the NBV position including trading of information gain and travel cost. Recent work by Holz et al. [21] suggests improvements for continuous data acquisition for 2D mapping in confined spaces such as rooms. In their approach a room will be fully observed before a position with higher information gain is considered as a candidate NBV position. Additionally, due to continuous data acquisition, NBV positions are constantly evaluated to avoid reaching a position which is already observed. In [22] an exploration algorithm is developed for 3D data acquisition. Frontier cells, defined as known cells adjacent to unobserved cells, are ranked according their length of connecting grid cells. A subset of these are then ranked according their travel cost with the one with the minimal cost is chose as the NBV position.

Our approach is closest to [2,18,20,19,1] in terms of generating a suitable set of potential new viewing positions and evaluating

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

the information gain. However, in contrast to previous work, we directly reason over the unknown space itself and estimate the expected knowledge gain from new views of cluttered environments. Previous proposed methods assume full visibility to the target region or object, which makes them poorly suited for cluttered environments since unseen objects in the occluded space lead to overestimation of the reward for potential new observation positions.

3. Next best view estimation

In the NBV problem we are given the task to sequentially observe an environment by moving a sensor to new viewing positions. The process seeks to minimize some aspect of unobserved space (*e.g.* its size), though the exact form of the objective function will govern the precise optimization being sought.

At each stage of the process, unobserved regions of the environment are estimated using an occupancy grid and a ray-traversal algorithm, shown in Fig. 1(a). Each cell c_n in the occupancy grid $\mathcal{O}$ is represented by a random variable $o_n \in [0,1]$ with $n = 1 \dots N$, where N is the total number of cells. The occupancy state $o_n = 0$ encodes the cell o_n as *free* and $o_n = 1$ as *occupied*. In order to estimate the NBV position many popular approaches [20,1,2,21] formulate the problem in information theoretic terms and evaluate the gain in information of a set of potential new viewing positions $S = \{s_1, s_2, \dots, s_k, \dots, s_K\}$. The NBV position $\hat{s} \in S$ is then estimated by maximizing the expected information gain:

$$\hat{s} = \arg\max_k E[I(s_k)]. \tag{1}$$

One information measure is entropy H, the amount of uncertainty in a discrete random variable. The entropy $H(o)$ expresses therefore the uncertainty about the state of a cell, in this setting whether the cell is free or occupied. For every potential new viewing position s_k, after integrating new measurements, the probability distribution of the occupancy grid either stays constant or changes. To measure the change we compute the information gain (Kullback–Leibler divergence) of the occupancy grid before and after integrating new measurements from a potential position s_k. Following the notation of [23], the information gain I over the occupancy probability $p(o_n)$ of a given cell c_n is defined as:

$$I(p(o_n \,|\, s_k, z_{n,k})) = H(p(o_n)) - H'(p(o_n \,|\, s_k, z_{n,k})), \tag{2}$$

where $p(o_n \,|\, s_k, z_{n,k})$ is the occupancy probability after integrating a new measurement $z_{n,k}$ from location s_k according to our sensor model and where $H'(p(o_n \,|\, s_k, z_{n,k}))$ is the posterior entropy.

To estimate the posterior entropy we need to know what measurements will be measured if the sensor is placed at a certain position s_k. However, since this is not possible due to the unobserved portions in the occupancy grid, we have to integrate over all possible measurements to compute the expected information gain for a viewpoint s_k:

$$E[I(s_k)] = \int_z p(z \,|\, \mathcal{M}, s_k) \sum_{o_i \in C(s_k,z)} I(p(o_i \,|\, s_k, z_{i,k}))\, \mathrm{d}z. \tag{3}$$

The set $C(s_k, z)$ defines the cells which are covered by the measurement z in the current occupancy grid map $\mathcal{M}$. Solving this integral in closed form is not feasible since we would have to integrate over all possible measurements.

A common approximation for the expected information gain is obtained by predicting the measurement z_k at position s_k using a ray-traversal algorithm. Typical methods to approximate the measurement include Markov Monte Carlo Sampling [23] or assuming the measurement to be the first occurrence of a cell with high occupancy probability [18,20], $p(o) > 0.5$. The latter approximation assumes infinite visibility until a cell with high occupancy probability is in line of sight. However, objects which have not yet been observed could potentially block the view to unobserved cells and therefore limit the actual information gain of a viewing position s_k. Thus, this could potentially lead to poor estimates over the unobserved space and the computed $\hat{s}$ is not necessarily the optimal choice, illustrated in Fig. 1(b). The better we predict the outcome of a potential new observation position s_k the more accurate we can choose $\hat{s}$.

We propose to approximate the expected information gain by predicting how likely it is to see a specific cell from a position s_k. That is, we estimate the probability that cell c_n is observable from position s_k. In detail, we define $p(x)$ to be the observation probability of a cell $c \in \mathcal{O}$. Let $x_n \in [0,1]$ be the random variable expressing the observation state, where $x_n = 0$ denotes the cell c_n is *not visible* and $x_n = 1$ denotes c_n is *visible*.

Given the observation probabilities we then estimate the expected information gain by estimating the posterior entropy over the occupancy state o of cell c, given x:

$$E[I(p(o \,|\, s, x, z))] \simeq H(p(o)) - H'(p(o \,|\, s, x, z)), \tag{4}$$

where z is the measurement predicted through a ray-traversal algorithm. We will now describe how the observation probabilities are computed in Section 3.1 and how they are used to estimate the posterior entropy is described in Section 3.2. Both of these subsections end with sample scenarios depicting the estimated distributions.

3.1. Observation probability

In the following, the assignment of the observation probability $p(x_n)$ is formalized. The observation probability depends on all cells

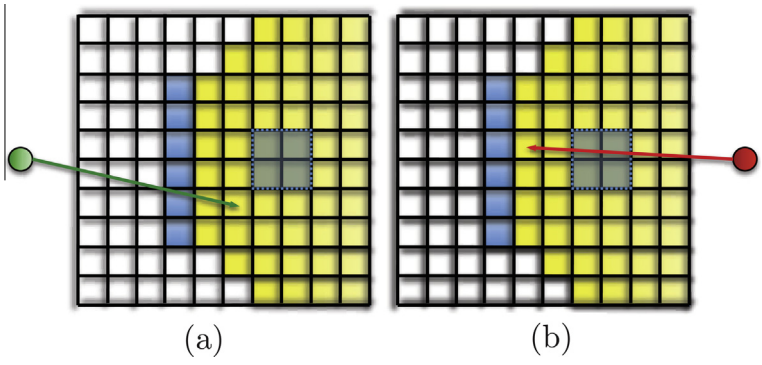

Fig. 1. The point cloud is represented as an occupancy grid. Blue cells are occupied space, white cells represent free space and yellow cells unknown space. Unknown space is estimated using a ray traversal algorithm shown in (a). To estimate the information gain of a potential next position we reason over the unknown space, as shown in (b). However objects which have not been seen yet (shaded square) lead to wrong estimations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

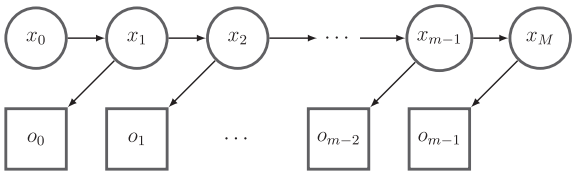

Fig. 2. The estimation process modeled as a Hidden Markov Model. The observation state x_m is estimated from state x_{m-1} given all prior occupancy probabilities up to o_{m-2}.

which lie in line of sight to the target cell c_n, estimated by a ray-traversal algorithm. Assume that the ray r to the target cell c_n penetrates M cells. Then r can be represented as the concatenation of penetrated cells $r = c_1 \dots c_m \dots c_M$. We want to estimate the observation probability of the target cell c_n given the cells in the ray. Intuitively, the observation probability of a cell $c_m \in r$ depends on the occupancy probability and the observability probability of the predecessor cell in the ray, $c_{m-1} \in r$. For instance, if the predecessor cell is highly likely to be occupied, the observation probability of c_m should be low. On the other hand, if the previous cell is likely to be free, chances are c_m is observable if the previous cell in the ray was also likely to be observable.

Thus we assume the observation probability to be Markovian and formulate the estimation process as a Hidden Markov Model, depicted in Fig. 2.

To estimate the observation state x_m we need to estimate the probability distribution associated with the current state conditioned on the measurements up to the state $m-2$. We can write the estimation of the probability distribution as a two step prediction-update recursion.

The prediction step, predicts the observation probability x_m from state x_{m-1} given all prior occupancy probabilities up to o_{m-2}:

$$p(x_m \,|\, o_{0:m-2}) = \sum_{x_m=\{0,1\}} p(x_m \,|\, x_{m-1})\, p(x_{m-1} \,|\, o_{0:m-2}), \tag{5}$$

where $p(x_m \,|\, x_{m-1})$ is the probability of transitioning from state x_{m-1} to x_m is. The state transition is the only open parameter in our system and controls the behavior of our approach, explained in more detail in Section 3.1.1. The prior observation probability of the first cell in the ray r_1 is set to $x_0 = 1.0$, stating full observability of the first cell $c_0 \in r$.

The update step to the cell c_m integrates the occupancy state of cell c_{m-1} and is defined as

$$p(x_m \,|\, o_{0:m-1}) = \eta\; p(o_{m-1} \,|\, x_m)\, p(x_m \,|\, o_{0:m-2}), \tag{6}$$

with η as the normalization factor. Furthermore the measurement $p(o_{m-1} \,|\, x_m)$ can be written in terms of the current occupancy state of cell o_{m-1}

$$p(o_{m-1} \,|\, x_m) \propto p(x_m \,|\, o_{m-1}) = 1 - p(o_{m-1}). \tag{7}$$

3.1.1. Visibility state transition probability

The observation probability of a cell depends on the occupancy probabilities of the cells in the path of the ray cast as well as the state transition probability of the Markov Process. The transition probability defines the likelihood of state transitions given their probability values. However, the transition probability also controls the behavior of the Markov process and with that the resulting observation probability of a cell. Hence, choosing different parameters for the state transition probabilities result in different visibility assumptions. This allows us to control the behavior of the observation probability when intersecting unobserved cells. As a result different parameters result in different NBV positions.

In the following we define the desired behavior of how cells in the ray should affect the target cell observation probability. Depending on the occupancy probability of the cells in the ray we want the observation probability of the target cell c_n to change in a certain way, when observed from a certain position s_n. If all cells in the ray r are likely to be free, we want the observation probability of cell c_n to be high (with high probability the cell c_n is observable). If the ray contains a 'likely occupied cell' at position c_m, we want the observation probability of the consecutive cell c_{m+1} to drop off steeply to a low observation probability value. Furthermore, we want the observation probability of all consecutive cells after c_m to be low and with that assign the target cell c_n a low observation probability. If the ray contains consecutive unobserved cells we want the observation probability to gradually fall. Assume the occupancy probability for the cell $\{c_0, c_1, \dots, c_{n-1}\}$is uncertain. Then we want the observation probability follow the rule:

$$p(x_1) > p(x_2) > \cdots > p(x_{n-1}) > p(x_n). \tag{8}$$

The more uncertain cells the ray needs to penetrate, the smaller is the probability of that target cell c_n being visible. Intuitively this expresses the fact that the more unobserved space a ray penetrates to get to cell c_n, the more likely it is that the ray will be blocked by an object that has not yet been observed.

To achieve this we make the state transition probability $p(x_{n+1} \,|\, x_n)$ to change adaptively depending on how far into the ray we are. Let the transition matrix be of the form

$$p(x_{n+1} \,|\, x_n) = \begin{array}{c} \\ vis \\ \neg vis \end{array} \begin{array}{c} \begin{array}{cc} vis & \neg vis \end{array} \\ \begin{bmatrix} t & 1-t \\ 0.1 & 0.9 \end{bmatrix} \end{array}. \tag{9}$$

Then the variable t controls how likely it is to stay in the state *visible* and with that how likely it is to transition to the state *not visible*. The farther we get into the ray, the less likely we want it to be to stay in state *visible*. Thus we make t dependent on the number of cells passed N_{cells} so far:

$$t = a^{N_{cells}}, \tag{10}$$

where a controls how steeply the value of t falls off.

3.1.2. Observability probability scenarios

Examples of how the observability probability evolves as we pass through the ray can be seen in Fig. 3. In the upper half of Fig. 3 we can see the evolution of the observation probability. This shows how the probability changes when the ray passes through the cells till it hits the target cell. In the lower part of the figure we can see four examples cases of what cells a ray might contain.

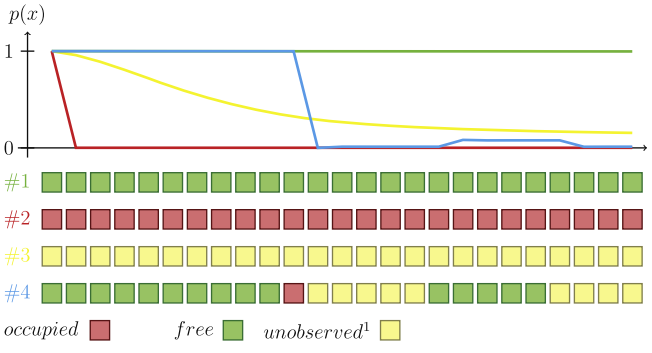

Fig. 3. The figure shows the change in observation probability when applied to our model. The upper half of the picture shows the evolution of the observation probability when traversing the cells in a ray. The lower part of the figure show the states of the cell in the ray.

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

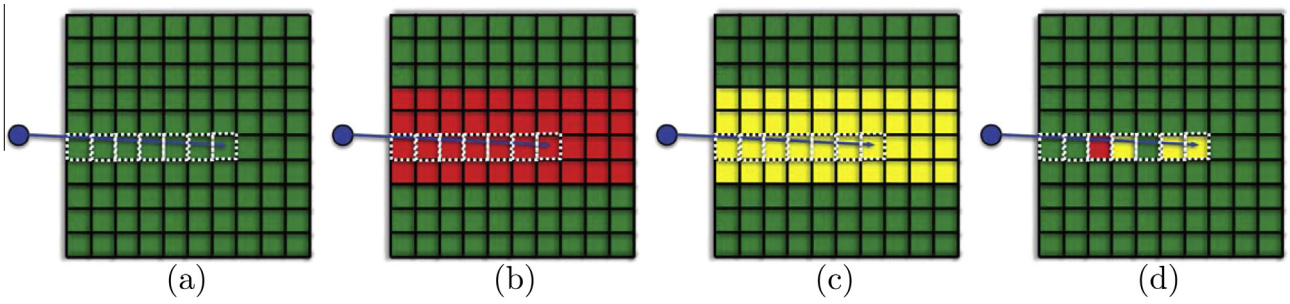

Fig. 4. In this figure we can see virtual scenes corresponding to the cases presented in Figs. 3 and 5. The blue circle on the left of all figures is the current sensor location, the arrow represents one ray shot into the grid to a target cell and the dashed white cells represent the cells penetrated by the ray. Green cells are free, red are occupied, and yellow cells are unobserved. In 4(a) only free cells are penetrated, in 4(b) only occupied cells, in 4(c) only unobserved cells and in 4(d) a mix of all are possible states are penetrated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Four different scenarios are shown, #1 where the ray-traversal algorithm had determined only free cells, #2 where only occupied cells where detected, #3 where only unobserved[1] cells where detected and finally #4 where a mix of all possible ray measurements where detected. In scenario #1 the observability probability stays almost constant close to $p(x) = 1$. This makes sense, since we do not pass any *occupied* or *unobserved* cells. In contrast, scenario #2 shows how $p(x)$ instantly drops to a value close to $p(x) = 0$ as soon as the first occupied cell is passed, and stays there since all new measurements are also occupied cells. The scenario #3 shows exactly the behavior that we have previously described. The ray consists of unobserved cells only, and the farther we get into the ray the more drops the observability probability $p(x)$. The last scenario #4 depicts the other wished for behavior, that is as soon as an occupied cell is passed, $p(x)$ drops and stays relatively small after that.

To better illustrate the cases and why the changes in probability occur in Figs. 3 and 5(a) virtual scenes is shown in Fig. 4. The sub figures correspond to the cases displayed in Fig. 3 and 5 (#1 is Fig. 4(a), #2 is Fig. 4(b), #3 is Fig. 4(c), #4 is Fig. 4(d)). Each cell in the grid is assigned a new observation probability, which depends on the cells traversed by the ray. The ray (blue error) is shot from the current sensor location (blue circle) to the target cell, with the white dashed boxes indicating all cell which are penetrated by the ray. The case #1 shown in Fig. 4(a) shows that the ray only traverses free cells. This occurs if we have estimated the cells in a previous observation as fee. In case #2 shown in Fig. 4(b) we traverse only occupied cells. This results when we have observed an obstacle so the cells occupied by the obstacle are occluded. In Fig. 4(b) which is shown in case #3 we traverse only unobserved cells. This occurs if the robot is positioned on the boundary of observed and unobserved space. In this case we have no knowledge about the environment and the probability behaves as shown in Fig. 3. In the last case #4 shown in Fig. 4(d) we traverse cells with different states. As we can see in Fig. 4(d) the ray first traverses free cells, followed by an occupied cell, followed by unobserved cell and so on. This occurs if we have partially estimated the probabilities of the cells but some of them are still unobserved. Since we penetrate an occupied cells, belonging to an observed object, the target observation probability immediately decreases to zero, since we can not observe the cell.

3.2. Posterior occupancy probabilities

So far we have explained how to compute the observability probabilities of cells. Now we describe how this observability of a cell is used to estimate the posterior occupancy probability $p(o_m \mid x_m, z_m)$. Intuitively, the occupancy probability now depends on two factors: how visible is cell c_m, which is captured by the observability state x_m and what is the occupancy state of the predecessor cell c_{m-1} in the ray r. Thus, we can again use a Bayes Filter to compute the posterior occupancy probability $p(o_m \mid x_m, z_{1:m})$, through a recursive prediction-update process. The prediction step is given through:

$$p(o_m \mid x_m, z_{1:m-1}) = \sum_{o_{m-1}=\{0,1\}} p(o_m \mid o_{m-1}, x_m)\, p(o_{m-1} \mid z_{1:m-1}), \tag{11}$$

where $z_{1:m}$ are again the measurements predicted through the ray-traversal algorithm. Note, how the transition matrix depends on the visibility state x_m of the next cell c_m. Intuitively, the visibility of a cell c_{m-1} should influence how certain the transition to the next state is. We will explain this in more depth in Section 3.2.1. The measurement update takes the usual form of

$$p(o_m \mid x_m, z_{1:m}) = \eta\, p(z_m \mid o_m)\, p(o_m \mid x_m, z_{1:m-1}), \tag{12}$$

where η normalizes the distribution.

3.2.1. Occupancy state transition matrix

The intuition of why the occupancy transition matrix should depend on the observability of the cell c_n becomes clear when we think about what effect the following transition matrix would have:

$$p(o_m \mid o_{m-1}) = \begin{matrix} \\ occ \\ free \end{matrix} \begin{matrix} \begin{matrix} occ & free \end{matrix} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \end{matrix}. \tag{13}$$

When using this transition matrix in the prediction step we will get a very uncertain belief $p(o_m \mid x_m, z_{1:m-1})$, even if the current occupancy state distribution $p(o_{m-1} \mid x_{m-1}, z_{1:m-1})$ was very certain. We would want this behavior if the observability of cell c_m is very low, because we want the occupancy distribution over c_m to become less certain. Only if cell c_m can be given a certain measurement update, the occupancy distributions becomes certain again. Thus we define the transition matrix depending on the observability probability as follows:

$$p(o_m \mid o_{m-1}) = \begin{matrix} \\ occ \\ free \end{matrix} \begin{matrix} \begin{matrix} occ & free \end{matrix} \\ \begin{bmatrix} t & 1-t \\ 1-t & t \end{bmatrix} \end{matrix}, \tag{14}$$

where

$$t = 0.5 + \frac{p(x_m)}{2}. \tag{15}$$

[1] Note, a cell is *unobserved* if its initial occupancy probability has not changed, meaning no measurement has been received for this cell yet. See Section 4.3 for more details.

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

Now, if cell c_m is likely to be visible the transition matrix is closer to the identity matrix and will not affect the posterior distribution $p(o_m|x_m, z_{1:m})$ by much. However, if cell c_m is likely to be not visible, the transition matrix will result in a less certain posterior distribution, unless a measurement with high certainty can be integrated.

3.2.2. Occupancy probability scenarios

We use the same scenarios from the observability estimation, to depict the behavior of the occupancy probability in Fig. 5 and illustrated as a virtual scene in Fig. 4. Again we can see how the resulting occupancy probabilities follow our intuition. Scenario #1 shows how the occupancy probability stays close $p(o) = 0$ as we pass through a ray full of free cells. Again in contrast, scenario #2 shows the opposite behavior, where the occupancy probability stays close to $p(o) = 1$as we pass only occupied cells in the ray. In scenario #3 $p(o)$ converges against the most uncertain state of $p(o) = 0.5$ when only unobserved cells are in the ray. Finally, scenario #4 depicts, how the occupancy probability becomes close to $p(o) = 1$ as soon as we observe an occupied cell, but then drops again to the uncertain state of $p(o) = 0.5$ after that when encountering unobserved cells, and only dropping to almost $p(o) = 0.0$ when measurements of free cells come in.

4. Preprocessing

In this section we describe the general framework of our approach. First we give an overview of the system architecture followed by a detailed description of the individual steps.

4.1. System architecture

Given a new point cloud $\mathcal{P}$, the first step is to perform the merging of two point clouds by transforming the new point cloud into the coordinate frame of the previous one. Let $\mathcal{P}_{n-1}$ denote the point cloud after $n-1$ cycles. Given the new point cloud $\mathcal{P}$, we join $\mathcal{P}_{n-1}$ with $\mathcal{P}$ to obtain $\mathcal{P}_n$. To merge two point clouds we need to find the rigid transformation of $\mathcal{P}$'s coordinate frame to $\mathcal{P}_{n-1}$'s coordinate frame. This transformation can be calculated using the current sensor pose and the first sensor observation pose. Using the current pose however leads to erroneous transformation matrices, due to inaccuracy in the sensor pose estimation. To improve the transformation we use an *Iterative Closest Point* (ICP) based approach. ICP is initialized with the estimated transformation using the current sensor pose and iteratively improves this initial transformation using information from $\mathcal{P}_n$ and $\mathcal{P}$. This procedure is explained in more detail in Section 4.2.

For further processing the point cloud $\mathcal{P}_n$ is converted into an occupancy grid. However, the sparse representation of the point cloud leads to holes in the occupancy grid representation of the scene. To fill these holes we use a Markov Random Field (MRF) in combination with iterated conditional modes (ICM) [24]. MRFs are undirected graphs with every node conditioned only on its neighboring nodes. ICM is a greedy method to sequentially maximize local conditional probabilities. Eliminating such holes leads to improved estimation of occluded space when using a ray-traversal algorithm for occlusion estimation. The MRF model optimization is described in Section 4.4.

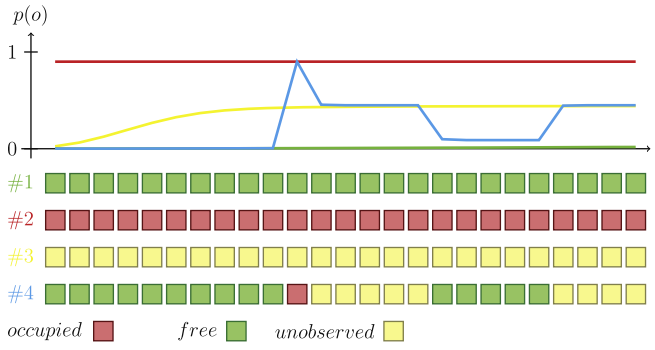

Fig. 5. The occupancy probability is shown for each cell in a ray. The upper half of the picture shows the evolution of the occupancy probability when traversing the cells in a ray. The lower part of the figure show the states of the cell in the ray.

In the next step we estimate the occluded space in the occupancy grid representation by using a ray-traversal algorithm. This also allows us to update the occupancy probabilities of every cell in the occupancy grid given our sensor model. The occupancy probability is either changing to 'likely free' or remains at uncertain. The definition of the occupancy grid as well as the ray-traversal algorithm used for estimating the occluded space is explained in Section 4.3.

The final step is to determine the next best viewing pose by choosing the pose that yields the highest expected information gain as explained in Section 3.

4.2. Point cloud registration

In practice, estimation of the current sensor pose will become inaccurate over time. When transforming the point clouds based on erroneous localization, the clouds will be affected by the same drift.

To deal with this problem we use ICP [25] to find the transformation that minimizes the difference between two point clouds. ICP is designed to fit points in a target point cloud to points in a control point cloud. For ICP to work we need an initial transformation to align the point clouds coarsely before ICP is applied. The erroneous rigid transformation, obtained through the current poses estimation, is used as the initial transformation.

4.3. Occupancy grid

To estimate and represent unknown space we use an occupancy grid, which is a discretized probabilistic representation of the environment. This makes it possible to assign every position in space a probability representing knowledge of the world and use efficient ray-traversal algorithm to estimate the unobserved space.

Every cell $c_{x,y,z}$ in the occupancy grid $\mathcal{O}$ is assigned a continuous random variable $p(o)$ denoting the occupancy probability of the cell. A probability of $p(o) = 1.0$ stating the cell is occupied while a probability of $p(o) = 0.0$ as free. An unobserved cell state is expressed with $p(o) = 0.5$.

Initially, all cells $c_{x,y,z} \in \mathcal{O}$ are initialized with probability 0.5, stating no knowledge about the state of any of the cells. For every new measurement $p \in \mathcal{P}$ we want to update the corresponding cell $c_{x,y,z} \in \mathcal{O}$ with a new occupancy probability. We update the occupancy grid using a stochastic sensor model defined by a probability density function $p(m|z)$, with m being the actual measurement and z being the true parameter state space value. In our case z corresponds to the cartesian coordinate of the cell center. We assume the probability function $p(m|z)$ to be Gaussian.

To determine the cell occupancy probability we incrementally update the occupancy grid using a sequential update formulation of the Bayes' theorem [26]. Given a current estimation of the occupancy probability $p(o)$, we want to estimate $p(c_{x,y,z}|m_{x,y,z})$, given a new measurement m. The update is formulated with

$$p(p(o)\,|\,m_{t+1}) = \frac{p(m_{t+1}\,|\,p(o))\;p(p(o)\,|\,m_t)}{\sum_{p(o)} p(m_{t+1}\,|\,z)\;p(z\,|\,m_t)}. \tag{16}$$

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

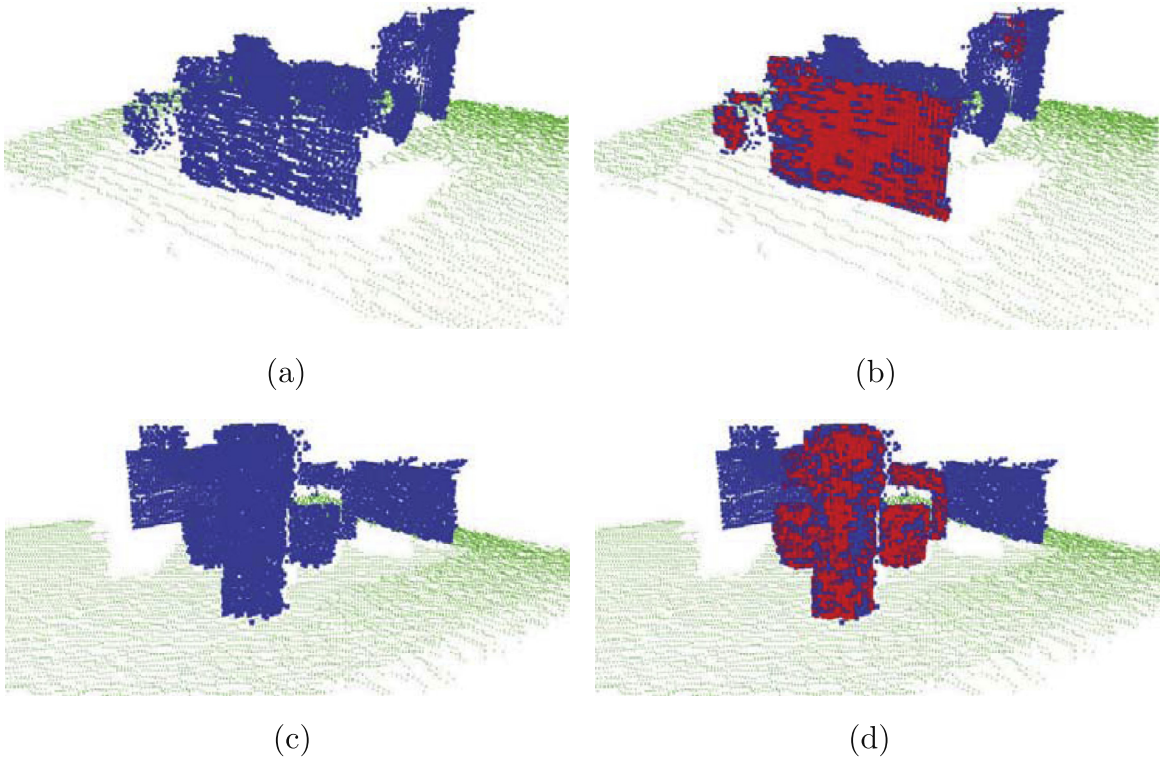

Fig. 6. In (a) and (c) we can see missing data of the box and a cylinder. This missing measurements lead to holes in the objects. The result of the MRF approach can be seen in (b) and (d) where the red cells are the cells set to occupied after convergence of the MRF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Furthermore, let $\mathcal{U}$ be the set of all cells which have a probability of $c_{x,y,z} = p(o) > 0.4 \wedge p(o) < 0.6 \in \mathcal{O}$.

4.3.1. Unknown space estimation

Point cloud data structures contain no information about the space where no measurements have been returned, consequently there is no distinction between space which is free and space which could not be measured due to occlusion. The distinction between the two types of space is important, because observing unobserved space gives us new scene information. With the use of a ray-traversal algorithm we distinguish between observed free space and occluded unobserved space. The ray-traversal algorithm we are using to estimate the new occupancy probability of the cells is describe in [27].

To estimate the occluded part of the scene we shoot rays from position P to every uncertain cell $c_{x,y,z} \in \mathcal{U}$. P is the position of the sensor from which it has acquired the input point cloud. The new probability of the cell depends on the ray ending in the cell $u_{x,y,z} \in \mathcal{U}$. Every ray r contains all cells it penetrates until it hits the target cell. If the ray contains a cell with probability $p(o_{x,y,z}) > 0.6$ (meaning the ray has penetrated a cell which is occupied) the target cell probability will remain the same. If the ray contains only free cells with $p(o_{x,y,z}) < 0.4$ or unobserved cells we change the probability of the cell according to our sensor model. This changes the occupancy probability of the cell towards more likely to be free. The change of the probability distribution is justified because P_i was the actual acquisition position and we can be certain that we see the unobserved cell if we penetrate only likely to be free or unobserved cells. Otherwise we would have measured an occupied cell and therefore penetrated an occupied cell.

4.3.2. Cost

Depending on the occupancy grid dimensions the number of cells can be very large. The cost for estimating the unobserved space from an acquisition position P depends on the number of unobserved cells $\mathcal{U}$ times the number of cells in each ray. Initially, the set $\mathcal{U}$ is fairly large since $\mathcal{O}$ is initialized with all cell probabilities set to 0.5 minus the cells that we have received measurements for and set to most likely to being occupied. However the number of unobserved cells drastically decreases with every new data acquisition, hence the cost for reestimating the unobserved space after incorporating new measurements also decreases.

4.4. Markov Random Field

The sparseness of point cloud can lead to holes in the surfaces of objects in the environment when converting them into an occupancy grid, resulting in cells that should be occupied remain unobserved. This phenomenon can be seen in Fig. 6(a) and (c), showing a partially observed box and cylinder. Holes in the objects lead to incorrect estimation of the unobserved space. Thus we want to identify the unobserved cells which belong to an object and set their probability to occupied. Intuitively, cells that are surrounded by occupied cells are likely to be part of the object as well. These correlations between neighboring cells can be captured using a Markov Random Field (MRF), were each variable only depends on the neighboring variables. In a $1D$ MRF the probability value of a random variable X_n depends on its two neighbors

$$P(X_n|X_{n-1}, X_{n+1}). \tag{17}$$

With the point cloud situated in $\mathbb{R}^3$ the MRF in a 3-D setting has $S = \{s_{x,y,z}\}$ which is the set of hidden nodes and $O = \{o_{x,y,z}\}$ the set of observation nodes, where $x = 1 \ldots X,\ y = 1 \ldots Y,\ z = 1 \ldots Z$, with $X,\ Y,\ Z$ being the dimensions of the occupancy grid. The nodes in S are related to one another over a neighboring system. For a fixed site s we define a neighborhood $N(s)$. To make it more concrete for the site $s_{x,y,z}$ the neighborhood is defined as: $N(s_{x,y,z}) = \{(x-1, y, z),$

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

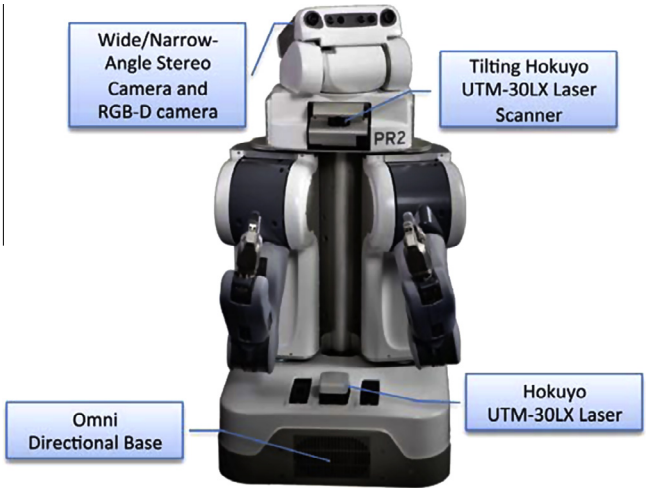

Fig. 7. The experimental platform (PR2 Willow Garage). In this paper, we use the tilting Hokuyo UTM-30LX laser scanner, mounted above the shoulder and an RGBD sensor mounted on the head, for data acquisition. (Image, courtesy of Willow Garage).

$(x+1,y,z),(x,y-1,z),(x,y+1,z),(x,y,z-1),(x,y,z+1)\}$. For efficiency we are not considering the diagonals. Thus in a 3D setting, the probability of a node s depends on its neighborhood,

$$P(s|N(s)). \tag{18}$$

Both the observation variables $o_{x,y,z} \in \{-1,+1\}$and the hidden variables $s_{x,y,z} \in \{-1,+1\}$are initialized with the state of the corresponding cell, where $+1$ represents an occupied cell and -1 the state unobserved. Furthermore the structure of our graphical model can be summarized by presence of two types of cliques. The first group of cliques are formed by $\{s_{x,y,z}, o_{x,y,z}\}$ which have an associated energy function modeling the relationship of the observation variable and the hidden variable. We choose an energy function which favors equal signs of $\{s_{x,y,z}\}$ and $\{o_{x,y,z}\}$ by giving a lower energy if both have the same sign and a higher energy when they have the opposite sign

$$E(s_{x,y,z}, o_{x,y,z}) = -\eta \cdot s_{x,y,z} \cdot o_{x,y,z}, \tag{19}$$

where η is a positive constant. The other group of cliques are formed by pairs of neighboring hidden variables, $s_{x,y,z}$ and $z_n \in N(s_{x,y,z})$. Here, we also want the energy to be low if both cells have the same sign and a high energy if they are different. We define the energy function as follows

$$E(s_{x,y,z}, z_n) = -\beta \cdot x_{x,y,z} \cdot z_n, \tag{20}$$

with β being a positive constant. Thus, the total energy function of our model is given by:

$$E(S,O) = -\beta \sum_{s \in S} \prod_{z_n \in N(s)} s \cdot z_n - \eta \sum_{x,y,z} s_{x,y,z} \cdot o_{x,y,z}. \tag{21}$$

To maximize the joint distribution of the MRF we use the *iterative conditional modes* (ICM) [28] algorithm, which "greedily" maximizes local conditional probabilities sequentially. The optimal assignment of the hidden variables $s \in S$ is found by iterating over the field, taking always one node variable $s_{x,y,z}$ at a time and evaluating the total energy for the two possible states $s_{x,y,z} = +1$ and $s_{x,y,z} = -1$ while keeping all other node variables fixed. After calculating the total energy for both states, we assign s the state that produced the lower energy. It should be noted, that we only iterate over hidden variables $s \in S$ that where initialized with state unknown. This is done for two reasons: On the one hand, we are certain about the assignment of a cell being occupied, thus we do not want the state of hidden variables, that were initialized with $s_{x,y,z} = o_{x,y,z} = +1$, to change. On the other hand, this leads to faster convergence of the algorithm. The algorithm has converged if we have visited every site $s \in S$ at least once and no changes to the variables have been made. A typical result of the MRF optimization step is shown in Fig. 6(b) and (d).

5. Experiments

We evaluate our formulation of the NBV problem in the task of large scale exploration and detailed data acquisition using a robotic system. The two evaluation scenarios differ in their size and level of clutter. To evaluate the detailed data acquisition capabilities we use the robotic system to explore a table top environment and acquire object information. The table top scene is extremely cluttered and unstructured resulting in many occlusions and the need to observe the table top from different positions. This environment analyzes the algorithm in the presence of clutter. In the second experiment, which is conducted only in simulation, we use our approach to explore large office environments. Office environments are in general less cluttered and consist of large rooms and long hallways, with mostly free areas. We compare and evaluate our approach to different frontier-based exploration algorithms. Besides the size and the clutter of the two environments, they differ also in terms of viewing scale of the sensor. For detailed exploration, like the observation of objects on a table top, the sensor viewing scale is in general far bigger than the environment. This is different for large scale exploration usually, the dimensions of a room or a hallway exceed the sensor viewing scale.

For experimental evaluation we use the PR2 robot Fig. 7, a research and development platform, from Willow Garage, operated using the Robot Operating System (ROS) [29]. The PR2 robot has a mobile base, with a footprint of 668×668 mm, two manipulators as well as several high-end sensors (*e.g.* tilt laser scanner, stereo cameras, RGB-D sensor, IMU). The robot has a substantial size, which means the robot can not reach every position in space the next best view algorithm estimates. The consequence of this is almost always time consuming, since unreachability can only be determined in imminent vicinity of an unexpected object due to either unnecessary robot motions or due to the need of recomputing a new next best view positions. We will explain in the following chapters how we deal with this issue and what effect this has on the information gain. The simulation experiments are performed using the robot simulator Gazebo. This simulator is able to replicate the PR2 robot with all its functionality in a physics simulated environment. In simulation we are utilizing the tilt laser scanner as well as a RGB-D camera for data acquisition.

In the following we evaluate our approach in simulation as well as on a real robotic system. First, we evaluate the detailed exploration of table top environment using a real robot. Second, we evaluate our approach in a large scale exploration environment of two office environments in a simulation environment.

5.1. Real robot experiments

To analyze the performance of our approach in small and cluttered environments, we simulate different cluttered table top scenes. The scenes, shown in Fig. 9, consists of different objects and configurations of the objects on the table, varying in different levels of complexity and clutter. The table is setup in a way that the robot has the ability to position itself anywhere around it, Fig. 8. To fully observe the table top and with that all the objects on it, we used the robot's mobile base to position the sensor around our region of interest. For the detailed data acquisition we are using a

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

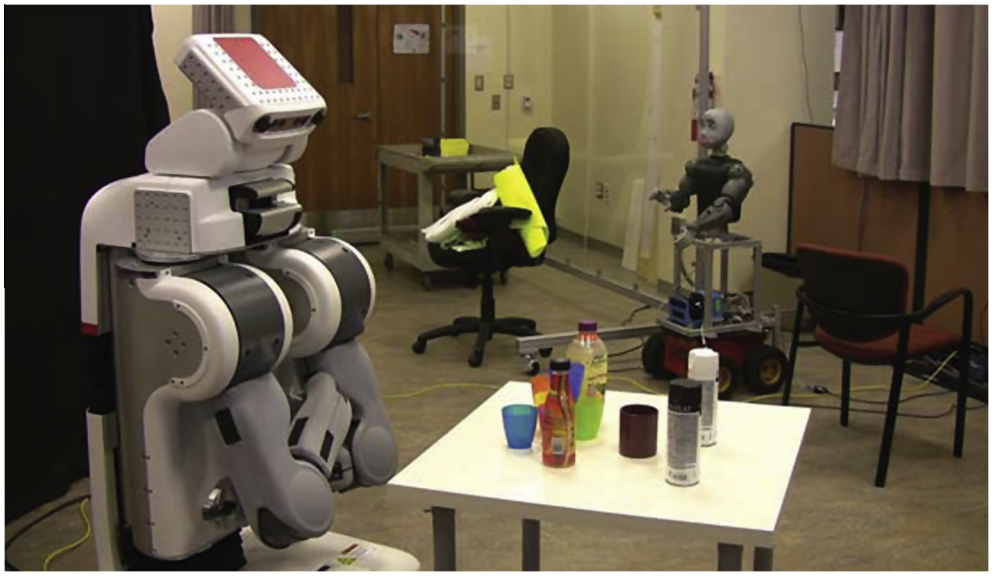

Fig. 8. PR2 robot acquiring data from a table top. The robot is able to position itself anywhere around the table.

tilting Hokuyo UTM-30LX laser scanner, mounted on the upper torso of the PR2 robot, to produce the input point clouds. The laser scanner is more accurate and less noisy compared to a RGB-D sensor, however it requires approximately 15 s to acquire a high resolution full 3D scan of the environment. Additionally, we restrict the lasers' field of view to 15° to increase accuracy. This makes an environment measurement costly in terms of needed time, therefore the goal is to keep the number of needed scans minimal and reduce the unobserved space maximally for every new scan. Furthermore, the laser scanner is mounted in a way that the robot cannot see all the objects by looking down at the table. This makes the setup more general and harder to solve because not all objects are seen in the first scan.

Since we are only interested in the objects on the table, we extract the objects from the table top as an additional pre-processing step. Given a new point cloud, the first step is to extract the table top and objects. The surface normals of all points in the point cloud are computed and are used to estimate all horizontal planar surfaces by using robust SAC (Sample Consensus) estimators. The approach is described in detail in [30] and implemented in the Point Cloud Library (PCL) [31]. All points above this planar surface correspond to data points of objects. We only integrate measurement points belonging to objects into the occupancy grid.

The dimension $(w \times d \times h)$ of the occupancy grid influences the resolution of the final measurement acquisition result as well as the runtime of the exploration algorithm. Given the relative small table top with 1 by 1 m, we choose the resolution of the occupancy grid as follows: $w = 129,\ d = 129$ and $h = 65$. The resolution of the occupancy grid influences the runtime of the algorithm greatly, since discretizing the space in smaller cells results in an increase of unobserved cells. More unobserved cells means more cells to traverse and process using the ray-traversal algorithm and with that an increases run-time. However, discretization size also influences the level of detail we observe the environment, in general we decided to have a high resolution for detailed observation while a low resolution is sufficient for observation of large space, for example. The resolution we picked for the detailed observation of the table-top results in a cell size smaller than 1 cm, which resulted in a very dense point cloud.

To execute the Markov Random Field preprocessing step to fill in missing measurements of the objects, the ICM algorithm requires the parameters η and β. They influence the convergence behavior and are set to $\beta = \eta = 1.0$ for a conservative convergence. The parameters influence the weight every relationship in a clique receives. Setting the parameters to $\beta = \eta = 1.0$ results in an equal weight. If we would set $\beta = 0$, we effectively remove the link between the paris of neighboring hidden variables. This would result in the global most probable solution is $s_{x,y,z} = o_{x,y,z}$.

Additionally the estimation of the NBV position depends on the parameter a, influencing how far we suspect to see into the unobserved regions. Best results were achieved with the parameter set to $a = 0.997$, which was empirically estimated.

In the detailed exploration experiments the NBV estimation depends on the number of possible sampling positions K and the angle θ of each s_k. We choose a finite set of viewing positions in such a way, that all of them lie on the perimeter of a circle around the region of interest. We have found that the discretization does not effect the result in a significant way, since small pose changes of the sensor do not effect the field of view in a notable way. However discretization drastically shortens the computation time. For our experiments, we have set $K = 10$ and θ_i with $i = [1, 2, 3]$ with the angles $[-15, 0, 15]$ resulting in 30 different viewing positions.

We compare our probabilistic framework with two other methods. The simple greedy approach assumes every unobserved cell is visible from a virtual scanning position, unless it is obstructed by an occupied cell. This predicted measurement z is then used to estimate and evaluate the information gain of a virtual scanning position $s_k \in \mathcal{S}$. This models the approach used in [20]. The second tested approach chooses the NBV position randomly after a new real measurement is taken by the sensor. To compare the three approaches we contrast the actual increase in information gain, which is equivalent to the decrease of unobserved cells in the occupancy grid. Additionally, we evaluate our approach in terms of prediction error of the information gain. A large prediction errors means overestimation of the information gain for a virtual viewing position. This has the affect that the chosen NBV positions is not necessarily a good one. Furthermore to give statical significance to our evaluation we conducted multiple runs for each scene and method. We conducted ten different runs for each scene and method, initialized from each of the sampling positions.

5.1.1. Evaluation

The overall performance shown in Fig. 10 shows the average decrease in unobserved cells over the 10 trials for each scan. We can

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

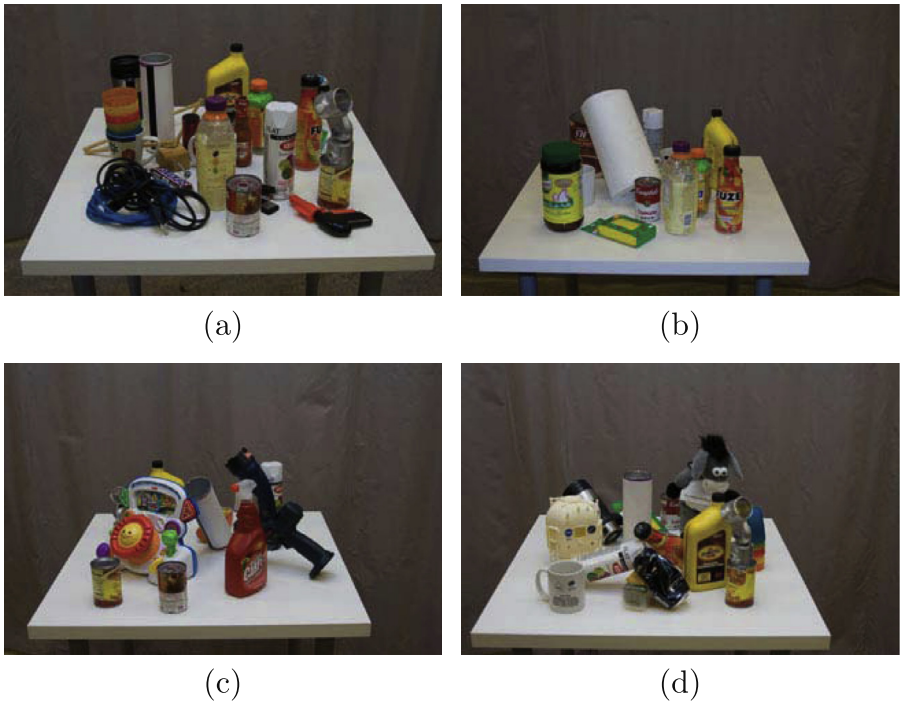

Fig. 9. In images (a)–(d) we show the experimental test scenarios for the detailed exploration. The table top scenes have different number of object simulating different level of clutter.

see that our proposed approach converges faster and has therefore reduced the number of unobserved cells faster than the simple greedy approach. This also means that to observe a certain percentage of the unobserved space, our approach can perform this in most cases much faster than the simple greedy method. As a consequence this means our method has chosen better next best scanning positions. An example of chosen positions and their reductions in unknown space can be seen in Fig. 12. Intuitively, the simple approach heavily over estimates the expected number of cells it can see from a virtual scan. As a result, it chooses next best scanning positions which decrease far less unobserved cells than expected due to objects, which have not yet been seen, blocking space which thought to be observable.

To chose the NBV position we have to predict the outcome of a virtual scan. In Fig. 11 we show that the absolute prediction error of our method is much lower than the prediction error of the simple method. The prediction error is calculated as the difference between the expected number of hitherto unobserved cells seen and the true number of hitherto unobserved cells seen in the new (real) scan. Note that due to the nature of the methods the two approaches computed two different sequences of sensor locations. Since this would result in an unequal comparison we ran the simple approach on the sequence estimated by our proposed method. Resulting in a comparison of the errors for the same scanning poses.

5.2. Simulated experiments

In the second experiment we apply our approach to two different large scale office environments, consisting of hallways and several rooms. Both environments are less cluttered, compared to the table top scene, but substantially larger. Due to the size of the environment the main difference to the first experiment, besides the level of clutter, is the fact that the maximal distance a sensor can obtain a measurement in, is generally significantly smaller than in the table top environment. The office environment as well as the robot is simulated using the robots simulator Gazebo. We constructed two large scale office environments shown in Figs. 13 and 14. The first environment consists of two adjacent large rooms with a total size of $15 \times 10 \times 2$ m. The second environment is larger in size ($20 \times 25 \times 2$ m) and consists of long hallways, small and large rooms. Furthermore, our approach is tested with different levels of clutter and varying starting locations of the robot. To explore the office environment we are using a simulated PR2 robot equipped with a simulated RGB-D camera (Microsoft Kinect). We limit the maximal sensor distance to 3.5 m, which is approximately the real maximal distance of the sensor. Additionally, Gaussian noise is added to the measurement by the simulator. Although the RGB-D sensor is capable of continuous data acquisition, we constrain our exploration such that the robot has to be stationary to obtain measurements. To fully observe the environment the robot needs to iteratively estimate NBV positions, position itself and the sensor at the new location and acquire new measurements until the space is observed.

Due to the significant larger environment we initialize the occupancy grid with a higher resolution of $w = 257$, $d = 257$ and $h = 65$. The parameter for model optimization, especially for the ICM algorithm, remain the same as in the first experiment. The visibility parameter a of our approach is set to 0.9999, resulting in little visibility penalties when looking into unobserved regions. In general, office environments consists of long and clutter free hallways as well as open spaces in offices.

Rather than relying on pre-determined or sampled potential new viewing positions we estimate new positions on a frontier-region. Frontier-regions are boundaries between observed free space and unobserved space. After a new point cloud is integrated into the occupancy grid, we estimate all boundary cells and cluster them to frontier-regions. Each centroid of a frontier-region is a potential new viewing position and will be evaluated for its expected information gain. On every NBV positions we capture a full 360° sweep of the environment and integrate the measurements into

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

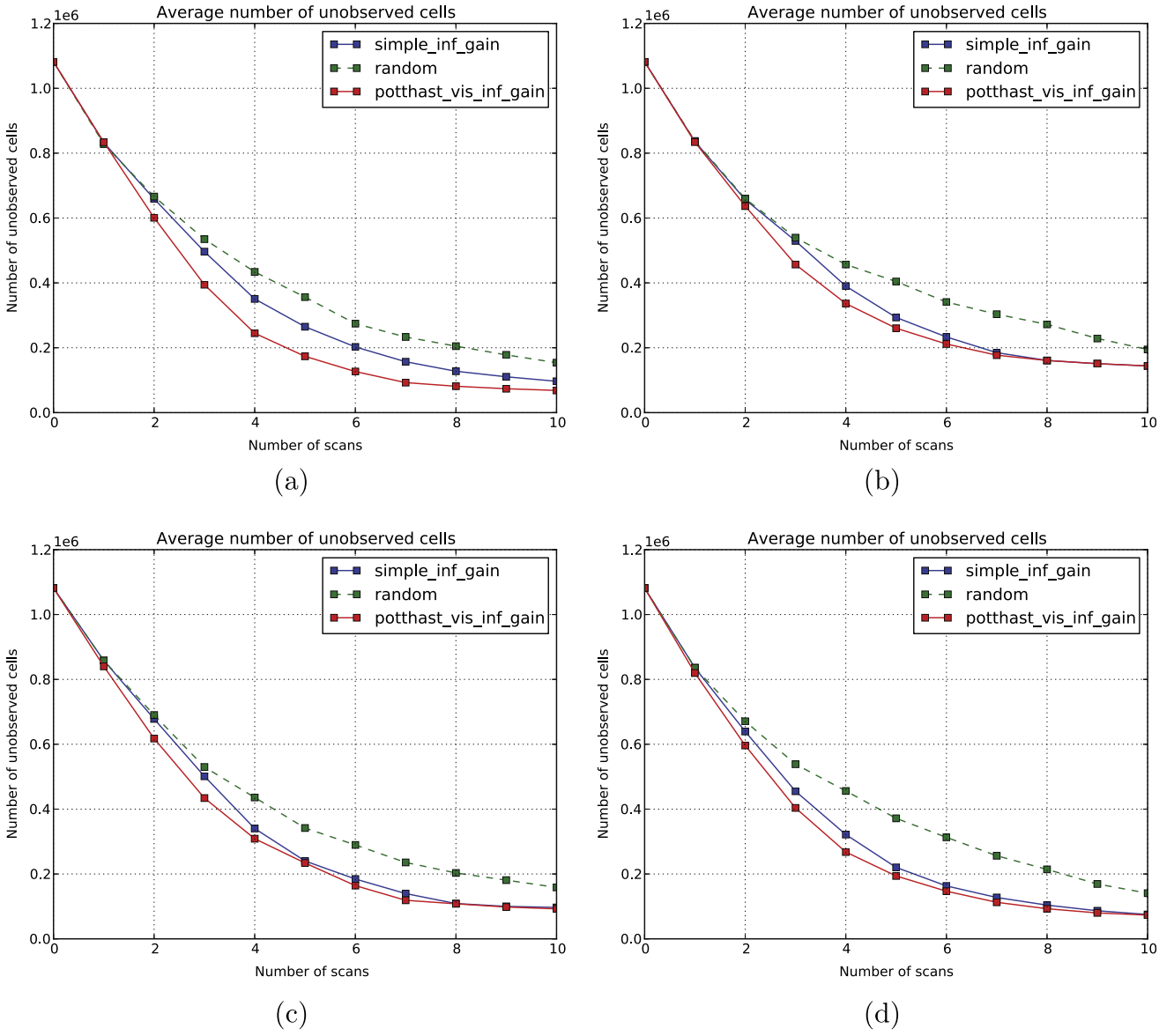

Fig. 10. In graphs (a)–(d) we compare the simple approach which is drawn as blue line with our proposed method drawn in red and a random approach in green. We can see that our method drawn in red decreases the number of unobserved cells faster, which is the effect of choosing NBV positions more optimal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the occupancy grid. Over the past decades many frontier-based approach were developed, the most prominent ones are [17,1] to which we compare our approach. Frontier approaches distinguish each other in the way NBV positions or frontiers are chosen and explored next.

The oldest but still widely used, because of it's simplicity, is Yamauchi [17]. The frontier centroid which is the closest to the robots current position is chosen as the NBV position. This strategy results in extremely short travel time after new measurements are acquired, however in general the number of necessary view points is of a factor of two larger compared to more sophisticated approaches. González-Baños and Latombe [1] developed a popular approach combining information theory and travel cost. For every frontier region the information gain is computed, with the measurement approximated by assuming infinite visibility, and penalized by the distance the robot would have to travel to get to that NBV position. The position with the highest reward is chosen as the NBV position and with that the next exploration position.

The robot navigation algorithm tries to navigate the robot to the chosen goal point. However, not all NBV positions are reachable by the robot, since the estimated centroids of frontiers could lie within or very close to objects. In this case the navigation algorithm tries to get the robot as close as possible to the goal point, but eventually has to aboard and proceeds as if the goal position is reached.

Furthermore, we compare our approach to the simple greedy approximation of the information gain as described in the previous experiment. The frontier centroid with the highest information gain is chosen to be the NBV position. Finally, we compare our approach to a random selection of the frontier centroids.

5.2.1. Evaluation

The performance for the different approaches in a large scale exploration task can be seen in Fig. 15 for the small office environment and Fig. 16 for the large office environment. As we can see, by setting the observation parameter close to 1.0 we converge to a

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

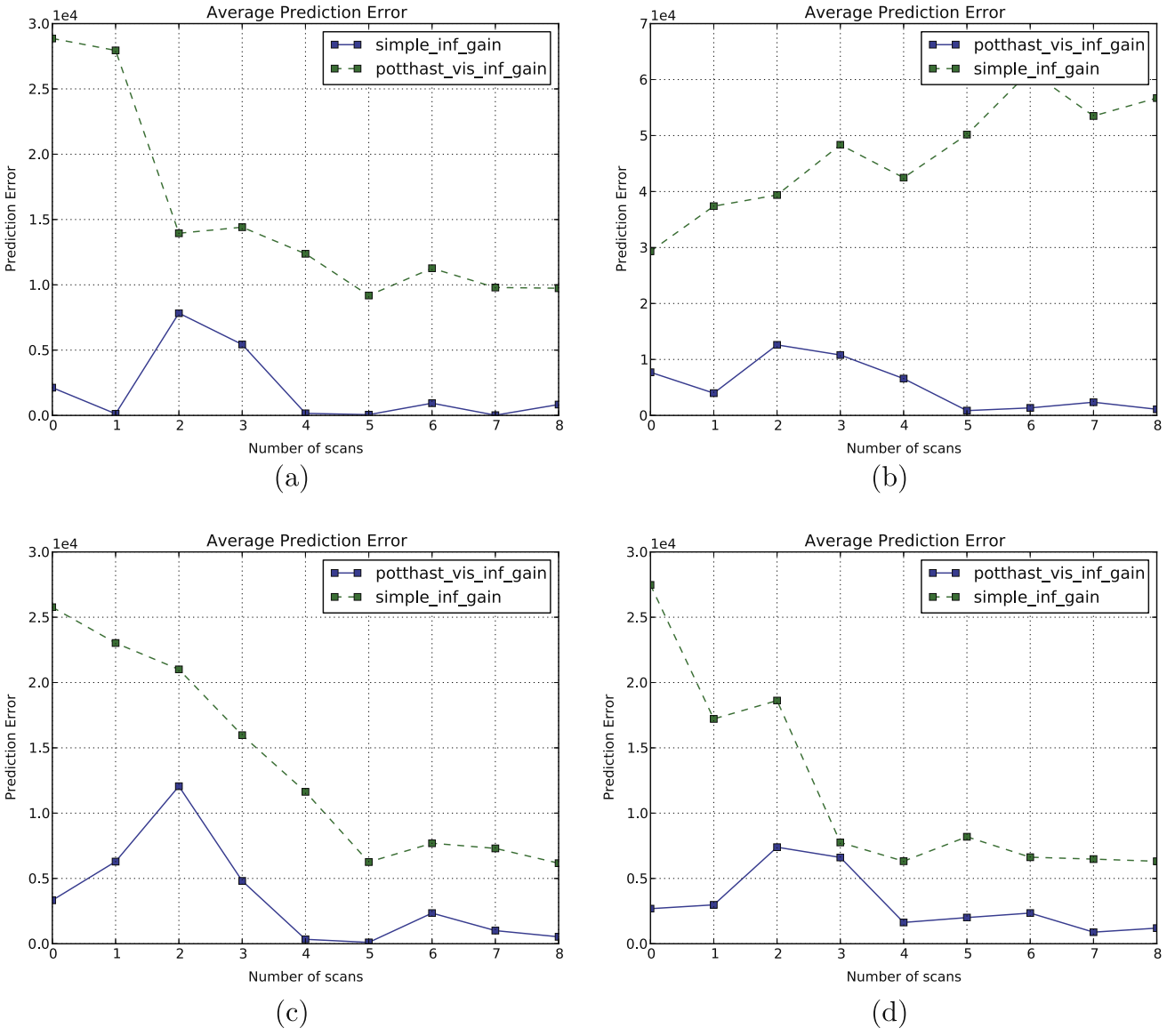

Fig. 11. In graphs (a)–(d) we show the absolute prediction error over the number of scans. The error is computed by taking the absolute error between the predicted information gain and the actual number of seen cells after a scan is taken. The green line in the graph represents the simple method while the blue line shows the prediction error of our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

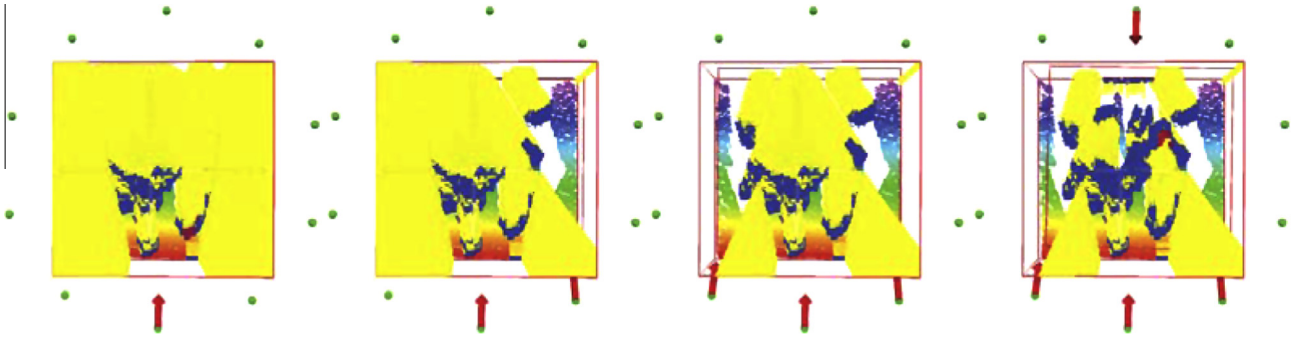

Fig. 12. The figure shows, from left to right, the reduction in unobserved cells after estimating the NBV position and acquiring new measurements.

similar behavior as the simple greedy information gain approach. Intuitively, it makes sense that due to the sparseness of the environment, the best exploration performance in terms of reducing the number of unobserved cells as quickly as possible, is achieved

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

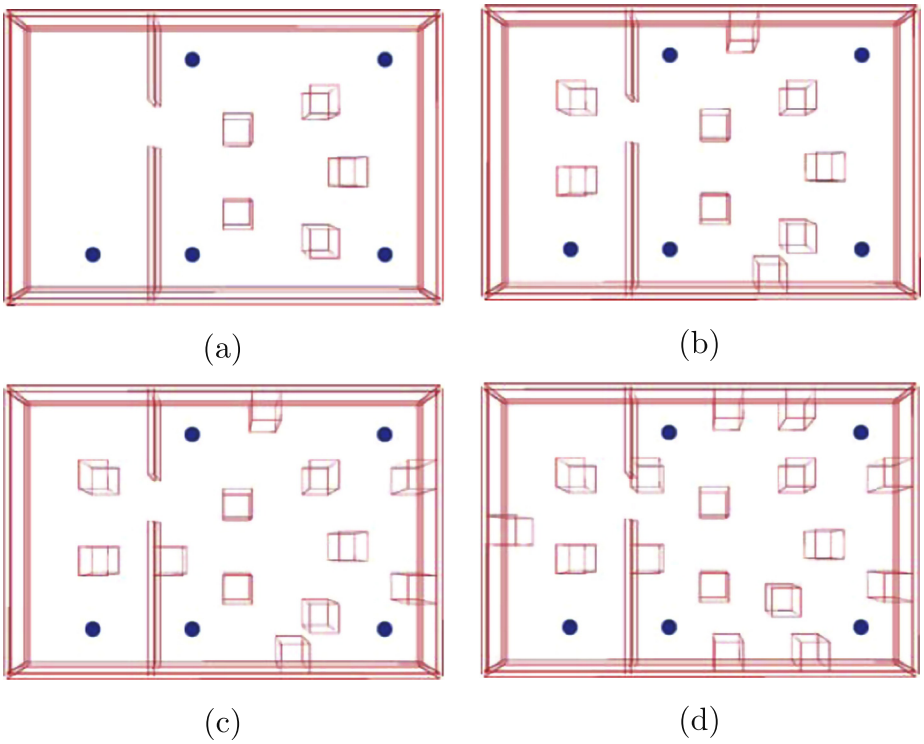

Fig. 13. The figure shows a small office environment, with different levels of clutter. The blue circles are the different robot starting locations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

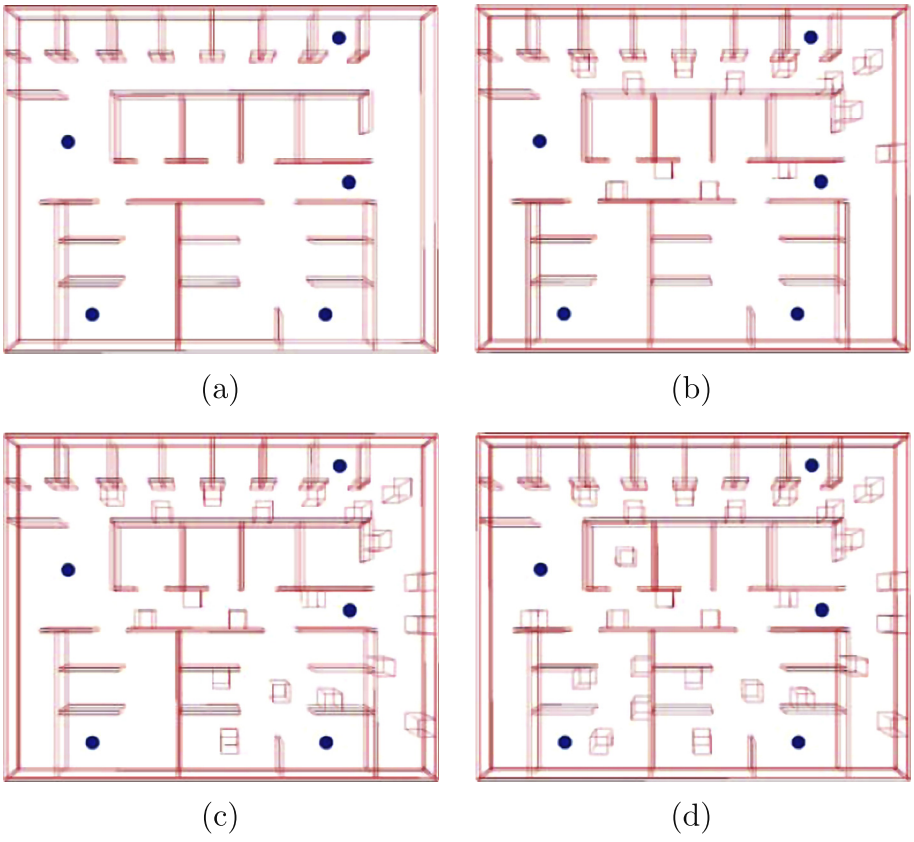

Fig. 14. The figure shows a large office environment, with different levels of clutter. The blue circles are five different initial robot starting locations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

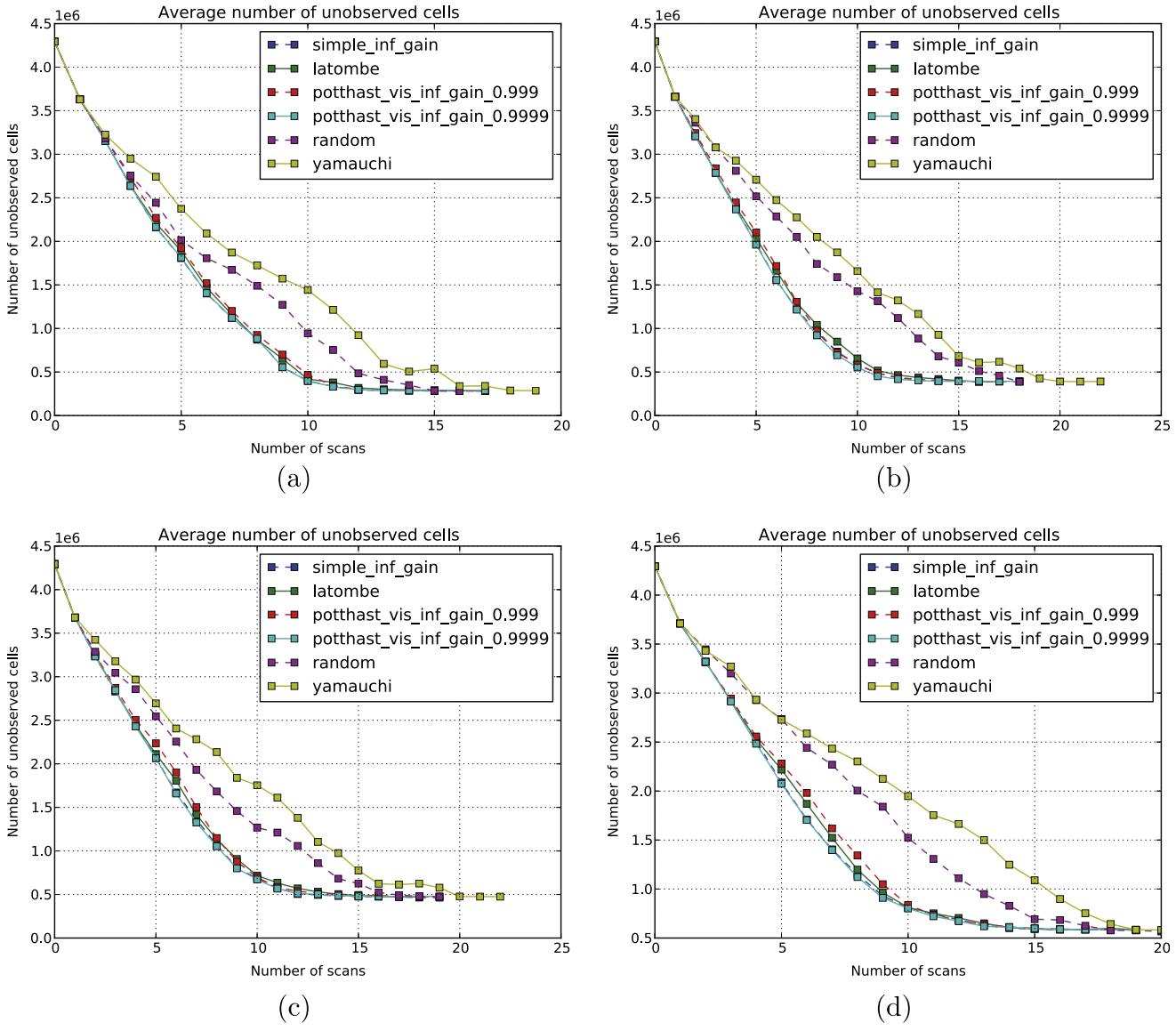

Fig. 15. The figure shows the average performance for the small office environment Fig. 13 of our approach compared to the simple greedy, Latombe, Yamauchi and random. The observability parameter for our approach is set to $a = 0.999$ and $a = 0.9999$. We can see that our approach is on par with all the other approaches.

by assuming almost infinite visibility. In both office environments this seems to be the best strategy, and our approach has the flexibility of adapting this rather greedy strategy. For both office environments experiments have shown that the performance for different values of the observation parameter does not degrade substantially. This is probably due to the fact that both environments are fairly large and sparse in obstacles. Hidden unobserved obstacles are rare and positions with large unobserved frontiers yield comparable performance.

5.3. Practical issues

The next best view algorithm we have presented in this paper is evaluated on a robotic system, more specifically on the PR2 robot. In general, the proposed next best view estimation algorithm is independent to a specific robot or any underlining system, however we use it to acquire data and more importantly get an estimated of the sensor positions in a word coordinate frame. In the following subsections we will give some insight into the practical issues when using our specific evaluation system.

5.3.1. Robot limitations

Depending on the system used to move the camera not all positions in space are reachable by the system. Our robot in particular is fairly large and can only operate on a 2D ground plane. Practically, this means potentially next best view positions can only be sampled from a subset of all possible viewpoints. This subset includes only the positions which lie within the robots configuration space. In contrast to our robot, a robot operating in 6D like an areal vehicle has a much larger operation space. However, even if we only sample in the robots configuration space, it is still possible that the positions sampled are not reachable. This happens if a potential viewing position lies within unobserved space and is obstructed by a previous unseen object. For the tabletop scene we assume there are no obstacles surrounding the table which could prevent the robot from reaching a sampled potential new viewing position. In the large environment exploration task this can how-

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

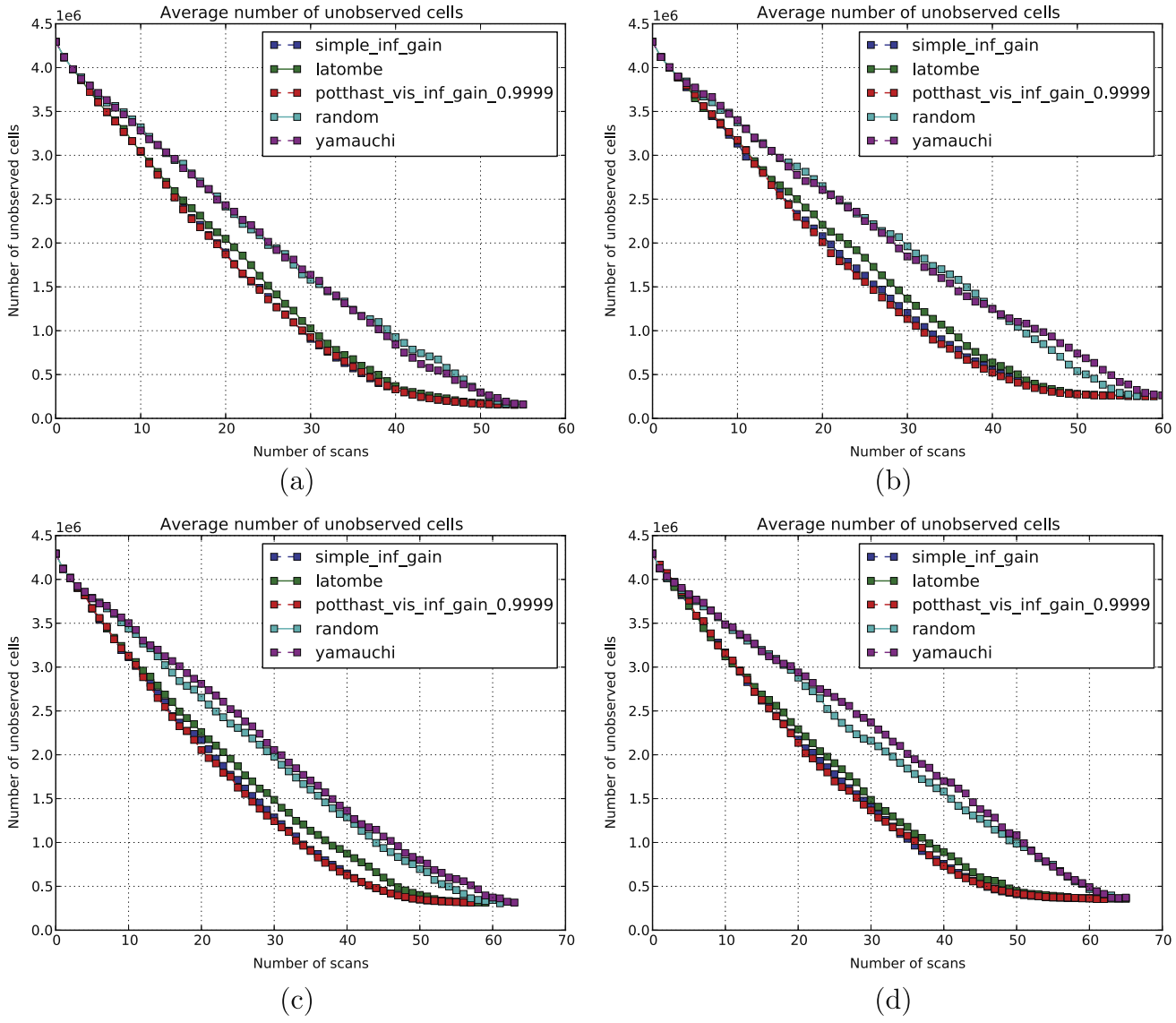

Fig. 16. The figure shows the average performance for the large office environment Fig. 14. Our approach is at least as good as the other tested approaches. The observability parameter is set to $a = 0.9999$.

ever happen, since the algorithm we use to estimate potential new viewing positions can return positions which lie within unobserved space. If such a position can not be reached a fallback plan has to be used. In our case we attempt to get as close as possible to the position, treating this as the new viewing position and acquire a new environment scan. However, there are many other operations that can be performed, *e.g.* choosing a different positions from the set of potential new viewing positions. Problematic with our fallback plan is that the estimated expected information gain, estimated by our proposed algorithm, for the fallback position is different than for the original position. This means taking a scan on the fallback position has a worse information gain than estimated on the original planned position or even as on a different position in the set of potential next best viewing positions. Unfortunately, we only know this once we have actually taken a new observation and integrated in our world representation. Moving to a different positions once the robot is not able to reach the position is a viable option, but it is not clear that this has more benefits. First, the robot has a lot more cost to actually reach a new positions, it first has to drive to the original, if not reachable, it has to drive to another one. Second, we do not know if this position is in fact better than the one close to the unreachable first one, since without taking an observation at the unreachable first position we have no information of its expected information gain.

5.3.2. Registration issues

Registration of sensor data over time is still an unsolved problem and a very active research field. In the real world experiment we have a tailored solution which works reasonable well and produces only small errors in registration. In the simulation environment, we do not assume perfect position knowledge of the robot, but rather apply gaussian noise to every position estimate. Registration errors accumulating over time will have an effect on the final representation of the world but also on the estimation of the expected information gain and with that on the choice of the next best view position.

5.3.3. Runtime

The runtime of the algorithm highly depends on the number of sampled potential new viewing positions and the total number of

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

cells that need to be evaluated. The runtime of estimating the expected information gain of one potential next best view positions therefore highly depends on the discretization of the environment. This varies from a second to a couple of seconds, due to costly ray-traversal operations. Currently we have a fast implementation, however there are GPU implementation which would improve the speed tremendously. Each additional potential next best view, which needs to be evaluated for its expected information gain, increases the total runtime by the evaluation time. Typically, the complete evaluation for the best next best view points takes about 30 s. A future goal would be to push this evaluation to under 10 s.

6. Discussion and future work

The results of the experiments show the usefulnesses of our approach in terms of detailed and large scale exploration. Comparing the results in the two different scenarios, one can see that reasoning over the observability of the cells is most useful in an extreme cluttered environment and when the sensor viewing scale is larger than the environment. In the detailed exploration setting our approach outperforms all other tested methods. On average, our proposed framework reduces the number of unobserved cells faster than the simple greedy method. In the large scale experiment the performance of our algorithm is at least as good as the simple greedy approach and is on par or better than the other tested approaches. We occasionally see a slightly faster reduction of unobserved cells using our approach, however in all of the cases it was insignificant.

When comparing our method to different approaches found in the literature the real advantage comes with that fact that we are able to mimic several different exploration strategies within one approach. Changing the value of the observability parameter a and with that the observability assumption of an unobserved cell also results in different exploration behaviors. This allows us to adapt our exploration strategy, depending on the task, within the same algorithm. With that our approach is more general to different exploration scenarios and needs. When changing the observability parameter to a value close to 1.0 we achieve no penalty for unobserved cells, resulting in the assumption of infinite visibility. In comparison when we set the parameter closer to 0.9 we achieve a behavior which mimics the behavior of always exploring the biggest frontier. Values in between 0.9 and 1.0 result in more or less penalization for unobserved cells.

A limitation of our approach as presented here is that the observability parameter a needs to be set manually, depending on what kind of scene needs to be explored. However, setting the parameter is very intuitive, for very cluttered scenes a value of $a \simeq 0.997$ is a good starting point. For large scale scenes with relatively few clutter, a value of $a \simeq 0.9999$ will most likely work well.

Future work involves building completely autonomous exploration system, that recognizes based on the task what observability strategy is the best. For instance, if the task is to first only explore a fully unknown map, a less stringent penalization should be used, to get a coarse exploration result. Then if the task is to focus into certain areas which are potentially are more cluttered, the observability parameter should be adapted to account for potentially unobserved smaller objects.

7. Conclusion

We have presented a flexible probabilistic framework for Next Best View estimation. We have shown that our approach can mimic several different explorations strategies. Thus the proposed framework is a more general approach to NBV estimation. This generality has been validated in several experiments conducted in simulation as well as on a real robotics platform. In both experimental settings, the detailed exploration and large scale office exploration, we can show that our approach is on par or better than other popular approaches like the simple greedy method.

References

[1] H.H. González-Baños, J.-C. Latombe, Navigation strategies for exploring indoor environments, Int. J. Rob. Res. 21 (10–11) (2002) 829–848.
[2] M. Krainin, B. Curless, D. Fox, Autonomous generation of complete 3D object models using next best view manipulation planning, in: Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2011, pp. 5031–5037.
[3] S. Haner, A. Heyden, Covariance propagation and next best view planning for 3d reconstruction, in: SSBA, 2012.
[4] E. Dunn, J.-M. Frahm, Next best view planning for active model improvement, in: British Machine Vision Conference, BMVC 2009, London, UK, September 7–10, 2009, Proceedings, British Machine Vision Association, 2009.
[5] C.I. Connolly, The determination of next best views, in: Proceedings of the 1985 Conference on Robotics and Automation, IEEE, 1985, p. 432.
[6] K.A. Tarabanis, P.K. Allen, R.Y. Tsai, A survey of sensor planning in computer vision, Trans. Rob. Automat. (1995) 86–104.
[7] W.R. Scott, G. Roth, J.-F. Rivest, View planning for automated three-dimensional object reconstruction and inspection, ACM Comput. Surv. (CSUR) 35 (1) (2003) 64–96.
[8] S. Thrun, Robotic mapping: a survey, in: G. Lakemeyer, B. Nebel (Eds.), Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002.
[9] A. Nüchter, H. Surmann, J. Hertzberg, Planning robot motion for 3d digitalization of indoor environments, in: International Conference on Advanced Robotics (ICRA), 2003, pp. 222–227.
[10] F. Amigoni, A. Gallo, A multi-objective exploration strategy for mobile robots, in: ICRA, IEEE, 2005, pp. 3850–3855.
[11] A. Hornung, B. Zeng, L. Kobbelt, Image selection for improved multi-view stereo, in: CVPR, 2008.
[12] R. Pito, A solution to the next best view problem for automated surface acquisition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (10) (1999) 1016–1030.
[13] N.A. Massios, R.B. Fisher, A best next view selection algorithm incorporating a quality criterion, in: Proc. BMVC, 1998, pp. 78.1–78.10.
[14] S. Haner, A. Heyden, Optimal view path planning for visual slam, in: Scandinavian Conference on Image, Analysis, 2011.
[15] M. Trummer, C. Munkelt, J. Denzler, Online next-best-view planning for accuracy optimization using an extended e-criterion, in: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 1642–1645.
[16] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, J. Denzler, An information theoretic approach for next best view planning in 3-d reconstruction, in: Proceedings of the 18th International Conference on Pattern Recognition, ICPR '06, vol. 1, IEEE Computer Society, Washington, DC, USA, 2006, pp. 103–106.
[17] B. Yamauchi, A frontier-based approach for autonomous exploration, in: Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation, 1997, pp. 146–151.
[18] P. Blaer, P.K. Allen, Data acquisition and view planning for 3-d modeling tasks, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 417–422.
[19] A.A. Makarenko, S.B. Williams, F. Bourgault, H.F. Durrant-Whyte, An experiment in integrated exploration, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2002, pp. 534–539.
[20] C. Stachniss, W. Burgard, Exploring unknown environments with mobile robots using coverage maps, in: Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI), 2003, pp. 1127–1132.
[21] D. Holz, N. Basilico, F. Amigoni, S. Behnke, Evaluating the efficiency of Frontier-based exploration strategies, in: Proceedings of the joint conference of the 41st International Symposium on Robotics (ISR 2010) and the Sixth German Conference on Robotics (ROBOTIK 2010), Munich, Germany, 2010, pp. 36–43.
[22] J. Wettach, K. Berns, Dynamic frontier-based exploration with a mobile indoor robot, in: Joint Conference of the 41st International Symposium on Robotics (ISR 2010) and the Sixth German Conference on Robotics (ROBOTIK 2010), 2010, pp. 1–8.
[23] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.
[24] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
[25] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 239–256.
[26] A. Elfes, Using occupancy grids for mobile robot perception and navigation, Computer 22 (1989) 46–57.
[27] J. Amanatides, A. Woo, A fast voxel traversal algorithm for ray tracing, in: Eurographics, vol. 87, 1987, pp. 3–10.

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006

[28] J. Kittler, J. Föglein, Contextual classification of multispectral pixel data, Image Vision Comput. 2 (1) (1984) 13–29.
[29] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, Ros: an open-source robot operating system, in: ICRA Workshop on Open Source Software, 2009.
[30] R.B. Rusu, A. Holzbach, M. Beetz, G. Bradski, Detecting and segmenting objects for mobile manipulation, in: ICCV, S3DV, Workshop, 2009.
[31] R.B. Rusu, S. Cousins, 3d is here: Point cloud library (PCL), in: IEEE Int'l Conf. on Robotics and Automation (ICRA), Shanghai, China, 2011.

Please cite this article in press as: C. Potthast, G.S. Sukhatme, A probabilistic framework for next best view estimation in a cluttered environment, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.07.006