2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

# Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning

Michael Krainin Brian Curless Dieter Fox

*Abstract*— **Recognizing and manipulating objects is an important task for mobile robots performing useful services in everyday environments. In this paper, we develop a system that enables a robot to grasp an object and to move it in front of its depth camera so as to build a 3D surface model of the object. We derive an information gain based variant of the next best view algorithm in order to determine how the manipulator should move the object in front of the camera. By considering occlusions caused by the robot manipulator, our technique also determines when and how the robot should re-grasp the object in order to build a complete model.**

## I. INTRODUCTION

The ability to recognize and manipulate objects is important for mobile robots performing tasks in everyday environments. Over the last years, various research groups have made substantial progress in recognition and manipulation of everyday objects [1], [2], [3], [4], [5]. While these techniques are often able to deal with noisy data and incomplete models, they still have limitations with respect to their usability in long term robot deployments because there is no provision for enabling a robot to learn about novel objects as it operates in its environment. This is an important limitation since no matter how extensive the training data, a robot might always be confronted with an unknown object when operating in everyday environments.

The goal of our work is to develop techniques that enable robots to autonomously acquire models of unknown objects. Ultimately, such a capability will allow robots to actively investigate their environments and learn about objects in an incremental way, adding more knowledge over time. In addition to shape and appearance information, object models could contain information such as weight, type, typical location, and grasp properties. Ultimately, such robots could become experts in their respective environments and share information with other robots, thereby allowing for rapid progress in robotic capabilities.

Toward this goal, we previously developed a technique that uses an RGB-D camera to track a robot's manipulator along with an unknown object grasped by the robot [6]. We used a reference design from PrimeSense [7] providing identical data to the XBox 360 Kinect [8]—$640 \times 480$ registered color and depth at 30Hz. The camera data was additionally used to generate a 3D surface model of the object. We showed that this approach can compensate for noise in manipulator motion and generate accurate models of household objects. However, this work has the key limitation that the manipulator motion has to be manually selected. Therefore, the robot is not able to autonomously build object models.


Fig. 1. Experimental platform. We used a WAM arm with BarrettHand on a Segway base. Mounted next to the arm on a pan-tilt unit is a depth camera.

In this paper, we overcome these limitations by developing an information-driven approach to 3D object modeling. Our system enables a robot to grasp an object and move it in front of its depth camera so as to build a complete surface model of the object. By considering occlusions caused by the robot manipulator, our technique can also determine when and how the robot has to re-grasp the object in order to build a complete model.

Our work provides the following contributions:

- We describe a probabilistic surface reconstruction approach that models the noise and sampling characteristics of depth cameras.
- Based on this reconstruction approach, we introduce an information gain based variant of the next best view algorithm that provides guidance for how to move the manipulator and how to re-grasp the object so as to minimize uncertainty in object shape.
- We show how to incorporate the effect of the robot manipulator on the object model and its visibility.

This paper is organized as follows. In the next section, we discuss related work and briefly review the system on which our work is based. Then, in Section III, we introduce our information-driven next best view approach. Our re-grasping strategy is described in Section IV, followed by experimental

Michael Krainin, Brian Curless, and Dieter Fox are with the University of Washington, Department of Computer Science & Engineering, Seattle, WA 98195. Dieter Fox is also with Intel Labs Seattle, Seattle, WA 98105.

This work was funded in part by an Intel grant, by ONR MURI grants N00014-07-1-0749 and N00014-09-1-1052, by the NSF under contract IIS-0812671, and through the Robotics Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement W911NF-10-2-0016.

results in Section V. We conclude in Section VI.

## II. RELATED WORK

The goal of next best view planning is to automate the process of view selection. It is a problem that has been considered by many researchers in the graphics and robotics communities. Next best view algorithms can be broadly divided according to the type of reasoning used (surface, volumetric, or global). We focus our discussion on volumetric techniques as they are most relevant to our work. For a good review of approaches, we refer the interested reader to the overview of Scott et al. [9].

Volumetric next best view algorithms typically involve some reasoning about which regions of space are empty, occupied, or unknown. Of particular interest are the boundaries between empty and unknown regions; these represent potentially viewable unknown regions. Connolly introduced the planetarium algorithm [10], which uses an octree to mark regions with labels. It simulates viewpoints sampled from the surface of a sphere and scores the viewpoints based on the solid angle of unknown area visible in the view. Massios and Fisher [11] take a similar approach but additionally consider the impact of new measurements on occupied regions by means of heuristics in their objective function. Our proposed algorithm is conceptually similar; however, our probabilistic and information theoretic interpretations lead us to a different functional form of the objective function.

Triebel et al. [12] also rely on information gain for next best view planning but using probabilistic occupancy grids instead of the signed-distance based grids we use in this paper. Signed-distance based grids have the advantage of explicitly constructing a function for use in extracting level set surface models; probabilistic occupancy grids do not have as clear a notion for surface reconstruction. Triebel et al. score viewpoints according to the information that would be gained during grid cell updates for the most likely measurement (as a stand-in for expected information gain, which is computationally prohibitive because it requires integrating over all possible measurements). Also notable in this work is that the authors optimize not only over a single next view location but over views along possible trajectories.

Because next best view planning aims to automate view selection, it is often used in conjunction with robotics, which can automate actuation to the viewpoint. In most cases, this takes the form of a robot moving itself to change the pose of a sensor [12], [13]. One exception is the in-hand modeling done by Zha et al. [14]. They use a robotic manipulator to move the object to the desired relative object-camera pose. In-hand modeling has the advantage of allowing the robot to explore areas of an object otherwise hidden by the table surface. However, Zha et al. do not consider the problem of changing grasp points to fill in holes left by the manipulator.

Performing object modeling requires knowing the aligning transformations between different sensor poses. The manipulator's pose should also be known for subtracting the manipulator from range images and for reasoning about potential occlusions. In previous work [6], we developed a system for articulated pose tracking and in-hand object modeling. The articulated pose estimation is essential because our manipulator can experience end-effector errors of multiple centimeters caused by cable stretch. The tracking is also important if such techniques are to be used in robots priced for consumer use. In this work, we use the Kalman-filter based tracking algorithm developed in [6] to provide hand and object poses.

## III. SINGLE GRASP VIEW GENERATION

The problem of generating views of objects held in a robotic manipulator differs from the typical next best view problem primarily in two ways. First, typical next best view scenarios involve an object resting on a surface such as a table with no other obstructing objects [11], [15]. The only hidden surfaces are those blocked by the table. During in-hand modeling, however, the manipulator may occlude substantial portions of the object in complex ways. There will be hand contacts that obstruct the scanner at all viewpoints and portions of the arm such as the wrist that block the object in view-dependent ways.

The second difference is the space of legal camera poses. Often in the next best view literature, the camera poses are pre-constrained based on the construction of a scanning rig [11], [16], or they are unconstrained by hardware but artificially constrained to lie on a surface such as a sphere for simplicity [10], [14]. In contrast, in-hand scanning lends itself to a more complex search space. Not all camera poses will lend themselves to inverse kinematics solutions or legal paths to those solutions. We therefore need to reason about the paths the manipulator might take.

In this section, we will assume the robot can achieve a first grasp of the object (for details, see Section V). We also assume that we have access to the relative poses of the object from frame-to-frame as well the poses, joint angles, and a 3D model of the manipulator. We use the output of a tracking and modeling system described in [6] to provide the necessary pose and joint angle estimates.

### A. Algorithm overview

Under the categories described in [9], our proposed next best view algorithm is a volumetric, generate and test approach. The volume we maintain stores weighted, signed-distance functions in voxels. At any point in time a maximum likelihood surface can be extracted via level sets. Additionally, voxels may be marked as *unknown* or *empty*, and the boundaries between them can be used for filling holes in the extracted surface.

We use the procedure in Alg. 1 to select new camera poses for partially modeled objects. Shown in line 1, the inputs are:

- $\mathcal{V}$, the volumetric model of the object
- $T_{hand}$, the pose of the manipulator in the coordinate system of $\mathcal{V}$
- $\theta$, the joint angles of the manipulator

In line 2, a mesh is extracted from the volume $\mathcal{V}$. Each vertex in the mesh has associated with it a confidence (or precision), the origins and usefulness of which are discussed

**Algorithm 1** Next best view of a grasped object

1: **procedure** SELECTVIEW($\mathcal{V}, T_{hand}, \theta$)
2: $\mathcal{S}_{obj}$ = ExtractMesh($\mathcal{V}$)
3: $\mathcal{S}_{hand}$ = GetHandModel($T_{hand}, \theta$)
4: $x_{obj}$ = GetObjectCenter($\mathcal{S}_{obj}$)
5: $dirs$ = GetPossibleViewingDirections()
6: **for** $dir$ **in** $dirs$ **do**
7: $range, roll, cost$ = SelectFreeDOFs($dir$,$x_{obj}$)
8: **if** $range, roll \neq \emptyset$ **then**
9: $x_{cam} = x_{obj} - range * dir$
10: $T_{cam} = (x_{cam}, dir, roll)$
11: $q$ = GetPoseQuality($T_{cam}$, $\mathcal{S}_{obj}$, $\mathcal{S}_{hand}$)
12: $score = q - \alpha_{cost} * cost$
13: **if** $q \geq t_q$ and $score > score^*$ **then**
14: $T^*_{cam}, q^*, score^* = T_{cam}, q, score$
15: **return** $T^*_{cam}, q^*$

below. In line 3, the manipulator model is set to the appropriate joint angles $\theta$ and is also transformed into the object's coordinate frame. These two lines produce the object and hand meshes $\mathcal{S}_{obj}$ and $\mathcal{S}_{hand}$ shown at the center of Fig. 2.

The next line determines the (approximate) center of the object. When planning views, we constrain the center of the object to lie in the center of the range image. This removes two translational degrees of freedom at very little cost. For the ranges at which our sensor operates (around 0.5 to 5 meters), objects held by the robot only take up a small fraction of the range image, so there is no concern of regions of interest being out of frame.

In line 5, possible viewing directions of the camera are selected. This results in a list of points on the unit 2-sphere (see Fig. 2 for an example) with some minimum angle $t_\theta$ to any previously used viewpoint. The role of $t_\theta$ is both to skip quality evaluations for previously used viewpoints and to allow view selection to "move on" if certain regions of the object cannot be imaged by the sensor due to properties of the material or geometry.

A viewing direction defines two of the rotational degrees of freedom of the sensor, and two of the translational degrees of freedom are constrained by the object center (in line 4). It remains to choose a distance from the object center and a roll about the viewing direction. While we could sample these degrees of freedom as well, that would greatly increase the number of pose quality computations we need to perform. Instead, we note that measurement qualities are typically better at shorter ranges and fix the range to a value around 0.7 meters unless no valid IK solutions can be found at this range. The roll, while relevant for stereo-shadow type effects, plays only a relatively minor role in the quality of a viewpoint. We therefore (in line 7) iterate through possible roll values and select the one with the lowest IK cost for some cost function on manipulator trajectories. In line 8, if no IK solution exists, the sample is discarded. The process of discarding non-achievable viewpoints is similar to the process used by Massios and Fisher [11].

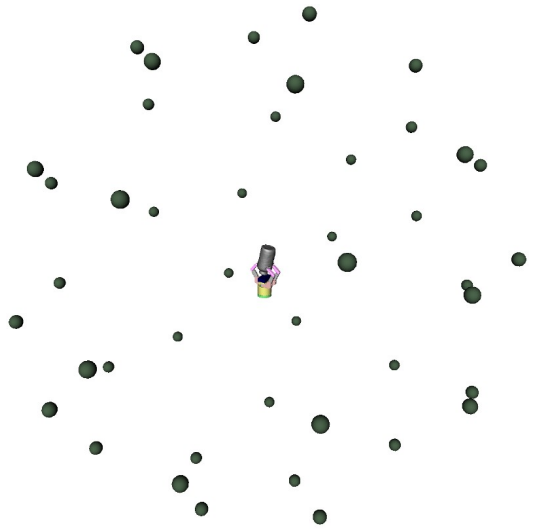


Fig. 2. Conceptual illustration of the view selection process. Viewpoints are sampled around the hand and object models. Each candidate camera pose is evaluated in terms of the view quality and the actuation cost.

In contrast to some existing next best view techniques [14], [16], [13], we do not require new views to overlap with existing ones. We receive sensor data along the entire trajectory and can perform registration and model update at multiple frames per second.

Given an achievable camera pose, as in line 10, defined by the camera location, the viewing direction, and the roll about the viewing direction, it remains to score the pose. GetPoseQuality in line 11 assigns a view quality to the pose according to the measure described below in Section III-B. Finally, in lines 12–14, we select the best camera pose according to a trade-off between view quality and motion cost among poses with quality above the termination threshold $t_q$. This encourages examining "nearby" regions first.

### *B. Statistical measure of pose quality*

We choose the volumetric method of Curless and Levoy [17] for representing knowledge, or lack of knowledge, about the object being scanned. This method is widely known as a tool for surface reconstruction. However, it also encodes statistical information about the uncertainty of each reconstructed surface point. We formulate a measure for viewpoints that will help in planning to minimize this uncertainty by approximately maximizing information gain.

First, we describe the probabilistic formulation for the volumetric method based on [18]. During scanning, we accumulate a set of registered depth maps $\{d_1, ..., d_t\}$. From these measurements, we can formulate a conditional probability given a possible reconstructed surface $S$,

$$p(d_1, ..., d_t|S), \tag{1}$$

and solve for the maximum likelihood surface. Under the assumption that depth samples are independent of one another, we can write the conditional probability as:

$$p(d_1, ..., d_t|S) = \prod_i \prod_{\{j,k\}} p(d_i[j,k]|S) \tag{2}$$

where $i$ indexes over depth maps, and $\{j,k\}$ index over pixels in a given depth map $d_i$. We assume that the uncertainty associated with a given depth sample $d_i[j,k]$ is one-dimensional and distributed along the line of sight for pixel $[j,k]$ in depth map $i$. Further assuming a Gaussian noise model, we set the per-pixel conditional probability to be:

$$p(d_i[j,k]|S) = \frac{1}{\sqrt{2\pi\sigma_i^2[j,k]}} e^{-\frac{(S_i[j,k]-d_i[j,k])^2}{2\sigma_i^2[j,k]}} \quad (3)$$

where $\sigma_i^2[j,k]$ is the variance along the line of sight of the pixel in depth map $d_i$ and $S_i[j,k]$ is the intersection of that line of sight with surface $S$. We can now formulate the negative log-likelihood function that is to be minimized:

$$L(S|d_1,...,d_t) = \sum_i \sum_{j,k} \frac{(S_i[j,k]-d_i[j,k])^2}{2\sigma_i^2[j,k]} \quad (4)$$

where we have omitted terms that do not depend on $S$ and thus do not affect estimation of the maximum likelihood surface. Given the density of measurements within a depth map, we can convert the pixel summation into an integral:

$$L(S|d_1,...,d_t) = \sum_i \int\int \frac{(S_i(u,v)-d_i(u,v))^2}{2\sigma_i^2(u,v)} dudv \quad (5)$$

where $(u,v)$ is the continuous image domain for each depth map. This objective can be minimized through the corresponding Euler-Lagrange equations. The maximum likelihood surface $S^*$ turns out to be the zero-crossing of the sum of signed-distance functions within a volume, where the signed-distance functions are distances from measurements along their lines of sight, weighted by the reciprocals of their variances.[1] See [18] for a detailed proof.

With this as motivation, we follow the method of Curless and Levoy [17] and construct a volume that stores sums of weighted signed-distances, as well as sums of weights. When extracting the surface corresponding to the zero level set, we also recover the weights across the surface. These weights, corresponding to the reciprocals of accumulated variances $\sigma_t^2(v)$, provide a measure of uncertainty over the surface at time $t$, stored at vertices $\{v\}$. The signed-distance functions in the volume are confined near the surface; behind them, the volume is marked unknown, and in front, along sensor lines of sight, the volume is marked empty. Extending the zero level set to include the boundaries between empty and unknown results in a complete surface, spanning measurement gaps. These added portions of the surface are assigned high variance, given that they are very uncertain, encoding only a bound on the surface and strongly indicating where more certainty is desired.

Given the current surface reconstruction, including empty-unknown boundaries, we can now formulate a metric for scoring potential new views of the surface in terms of reduction of uncertainty. Consider a new viewpoint, indexed by $z$. We can simulate the variance $\sigma_z^2[j,k]$ of a new measurement along line of sight $[j,k]$ by sampling the current surface estimate $S^*$ (generating a virtual depth map $d_z$) and computing per-pixel variance for this virtual view [18]. To measure the reduction in uncertainty, we also require the probability distribution along the hypothesized sensor line of sight for the current surface reconstruction. We approximate that the uncertainty in the reconstruction along the hypothesized line of sight $[j,k]$ is Gaussian with variance $\sigma_t^2[j,k]$.

We can now model the information gain of this measurement in terms of entropy reduction. We first note that updating accumulated variance from $t$ to $t+1$ given a new measurement $z$ takes the form:

$$(\sigma_{t+1}^2)^{-1} = (\sigma_t^2)^{-1} + (\sigma_z^2)^{-1} \quad (6)$$

The entropy of a univariate Gaussian distribution is $\frac{1}{2}\ln(2\pi e\sigma^2)$, which yields an information gain (difference of entropy for Bayesian updates):

$$G_{t+1} = \frac{1}{2}\ln(2\pi e\sigma_t^2) - \frac{1}{2}\ln(2\pi e\sigma_{t+1}^2) \quad (7)$$

$$= \frac{1}{2}\ln\left(\frac{\sigma_t^2}{\sigma_{t+1}^2}\right) \overset{(6)}{=} \frac{1}{2}\ln\left(1+\frac{\sigma_t^2}{\sigma_z^2}\right) \quad (8)$$

In the context of information gain of a new viewpoint $z$ relative to the current reconstruction $S^*$, we sum over the information gains across all lines of sight:

$$G_{t+1}(z) = \frac{1}{2}\sum_{j,k}\ln\left(1+\frac{\sigma_t^2[j,k]}{\sigma_z^2[j,k]}\right)\frac{d_z^2[j,k]}{f^2|n_z[j,k]|} \quad (9)$$

where $d_z[j,k]$ is the depth for pixel $(j,k)$ in the virtual depth map from viewpoint $z$, $f$ is the focal length of the sensor, and $n_z[j,k]$ is the $z$-component of the normal at that pixel. The factor $\frac{d_z^2[j,k]}{f^2|n_z[j,k]|}$ is the differential area on the surface of the model; this factor is needed for the summation in depth map space to correspond to area-proportional summation over the surface $S^*$.

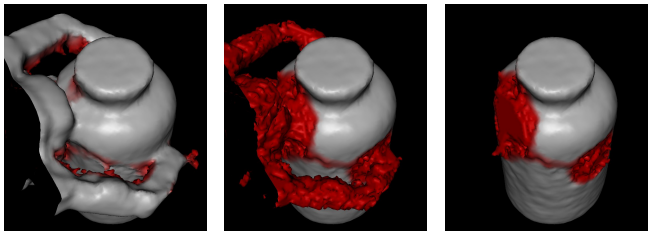
We do not include any pixel (with corresponding ray) that meets any one of the following criteria:

- The ray does not intersect $\mathcal{S}_{obj}$.
- The ray intersects $\mathcal{S}_{hand}$ before reaching $\mathcal{S}_{obj}$.
- The intersection with $\mathcal{S}_{obj}$ occurs within a distance of $t_{hand}$ to $\mathcal{S}_{hand}$ (would not be used for model update).
- The angle between the intersection's normal and the ray is too large (the sensor would not get a measurement).
- A ray from the projector component of our sensor [7] to the first ray's intersection is occluded or at too large an angle to the surface (accounts for some stereo effects).

The function GetPoseQuality in line 11 of Alg. 1 then simply returns $G_{t+1}(z)$ from (9).[2] In line 15, SelectView returns a camera pose and a quality. If the quality is below the threshold $t_q$, then the imaging is deemed complete for the current grasp, and the object is set back down. Otherwise, the robot moves to its IK solution for the best view and

[1] In practice, the weights reflect greater uncertainty where range sensors return data that is typically poorest in approximating the surface: near depth discontinuities and along grazing angle views of surfaces.

[2] Although the quality computation is performed on a mesh, we still consider the algorithm to be a volumetric next best view algorithm because the underlying structure is a voxel grid, and operations on the mesh could equivalently be done using volumetric rendering techniques.



(a) (b) (c)

Fig. 3. (a) Pill bottle when all measurements are used for reconstruction; (b) hand measurements used for carving only, not for adding signed-distance functions; (c) carving only for hand measurements and explicitly marking voxels within the hand as empty. Grey for high confidence, red for unknown boundary, and shades inbetween for low confidence.

updates the object model with range images from along the trajectory. The cycle continues until the threshold is reached.

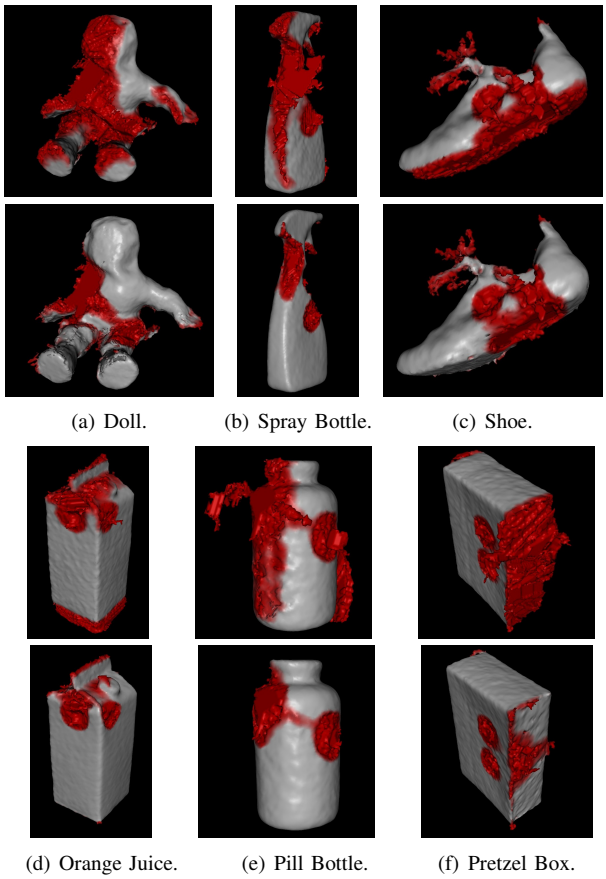
## IV. MULTIPLE GRASPS

After the object is placed back on the table the robot must decide whether and where to grasp the object to observe still unknown regions. Two necessary components for this reasoning are the identification of which object regions require more sensing and a model suitable for grasp planning. To achieve these goals, the object model must express uncertainty about any unknown regions under the hand and be free of protrusions where the hand was grasping.

Naïvely including all measurements in the depth maps into the volumetric model results in the manipulator surfaces becoming part of the object model (see Fig. 3(a)), which is unsuitable both for reasoning about uncertainty below the fingers and for grasping. To address these problems, we first use the known manipulator configuration to exclude measurements near the hand from adding surfaces to the object. That is, we use measurements within $t_{hand}$ of the manipulator only for marking voxels empty, not for adding signed-distance functions. The result (Fig. 3(b)) expresses uncertainty about regions hidden by the hand surfaces, but contains protrusions unsuitable for grasp planning. We additionally mark any voxels within the known hand volume as empty resulting in Fig. 3(c).

Grasp generation now becomes quite straight-forward. We require a grasp planner that given a mesh model can produce multiple stable candidate grasps. In our implementation, we generate candidate grasps using OpenRAVE [19]. We then run the procedure in Alg. 1 separately on multiple candidate grasps and select the grasp with highest quality. If no grasp has quality at least $t_q$, the procedure terminates. If a good grasp stability metric is available, one could instead select the grasp based on a weighted sum of quality and stability.

This algorithm has the benefit that it requires very little beyond what has already been proposed in Section III. By requiring the new grasp to have a high-quality view, we guarantee that the model will be more complete after the grasp. As future work, we also wish to explore approaches that explicitly attempt to minimize the number of grasps needed for complete coverage.



(a) Doll. (b) Spray Bottle. (c) Shoe.

(d) Orange Juice. (e) Pill Bottle. (f) Pretzel Box.

Fig. 4. Reconstructed models of objects grasped only once. For each object (a)-(f), we show the model after an initial trajectory only (top) and after the next best view procedure (bottom).

## V. RESULTS

For generating initial grasps we make use of a heuristic grasp planner written by Peter Brook, based on the findings of [20]. The object grasp point and approach direction are determined by first subtracting the table plane from the depth camera data and then computing the principal components of the point cloud representing the object. The approach is taken perpendicular to the principal axis.

Our next best view algorithm requires a few parameters. We set the confidences of hole-filled regions to correspond to a standard deviation of 5 cm. For termination of the procedure, we use a threshold of $t_q = 0.005$, and we use an angular threshold of $t_\theta = 10°$. We implement $cost$ in Alg. 1, line 7 as the estimated time for the trajectory and use $\alpha_{cost} = 0.0005$. To generate the models shown in the next subsection, we start with a simple initial trajectory and then compute views until $t_q$ is reached.

In all of the results that follow, red areas indicate empty/unknown boundaries; no observations of these surfaces have been made. Grey regions have high confidence, which occurs only after many observations. Shades inbetween indicate surfaces which have been observed but are of lower confidence.

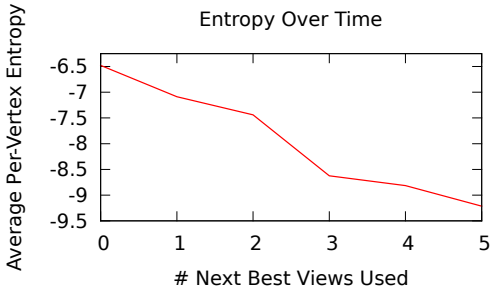



Fig. 5. Per-vertex entropy after each view selection for the orange juice bottle in Fig. 4(d).

### A. *View Selection*

Fig. 4 illustrates the results of our next best view algorithm for a single grasp on multiple objects. Next best view planning covers more of the empty/unknown boundary, resulting in more complete models. In Fig. 5, we take a more detailed look at the orange juice bottle example from Fig. 4(d). This figure plots the average per-vertex entropy of the model as a function of number of views used. The entropy decreases as the low confidence regions (red in Fig. 4) are replaced with higher confidence regions (grey).

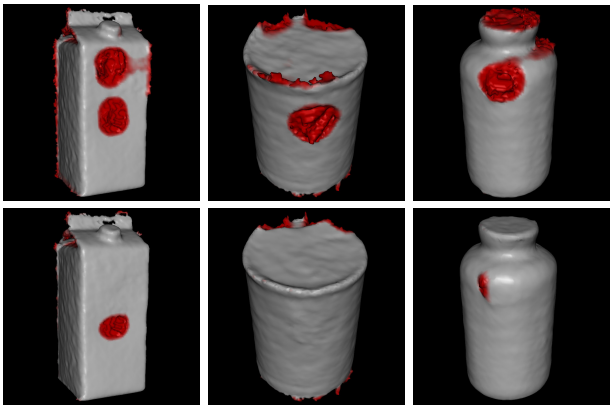
As currently implemented viewpoint evaluations can be performed around three times per second on an Intel Core i7 processor. The bulk of this time is spent in GetPoseQuality (Alg. 1, line 11), which performs software ray-casting sequentially per pixel. A more optimized implementation would instead implement GetPoseQuality using shaders, allowing for parallel evaluations of per-pixel information gain.

### B. *Grasp Selection*

We start by demonstrating how the grasp selection procedure applies in an example using a cracker box. After performing the next best view procedure for a single grasp, we apply the grasp selection procedure to the model, which is shown in Fig. 6(a). Note the regions of low confidence shown in red. We generate a grasp table using OpenRAVE and evaluate the top 10 legal grasps that it finds; generating these candidate grasps takes approximately one minute. The resulting grasps are shown in Fig. 6(b) sorted according to the view quality. Notice that the first grasps leave visible much of the region previously under the fingers and palm. The later grasps cover up much of this area.

In Fig. 7 we show examples of objects after two grasps. The next best view regrasping procedure allows previously occluded surfaces to be observed; however, more than two grasps may be required for complete coverage as there can be overlap of the occluded regions.

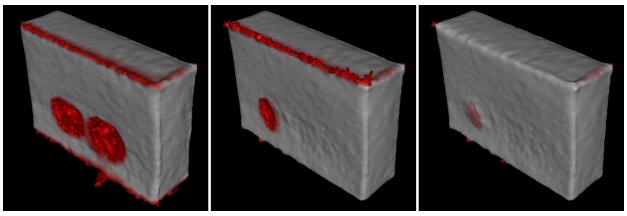
The process of regrasping introduces possible failure cases in the forms of tracking failures and grasping failures. The first case occurs if there is some large, unexpected motion of the object (e.g., falling over). While the tracking algorithm can handle smaller motions such as wobble, these larger motions still cause problems. The second case occurs if the grasp is not sufficiently stable. The problems of tracking and stable grasp generation are far from solved, but as they improve, so will our system.



(a) Orange Juice. (b) Coffee Can. (c) Pill Bottle.

Fig. 7. Object models after one grasp (top) and two grasps (bottom).

(a) Three grasps used to model a cracker box.



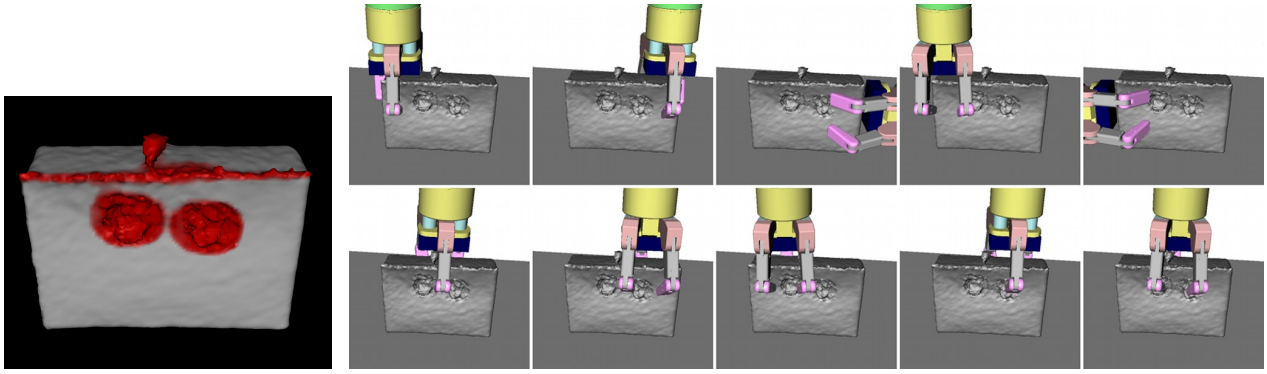
(b) Corresponding confidences after each grasp.

(c) Resulting meshed surfel model.

Fig. 8. Complete object model generated using three separate grasps.

Because of the potential problems that can occur during regrasping, we have mainly demonstrated our next best view algorithm with either one or two grasps, but holes may still exist. Fig. 8 shows a box modeled using three grasps, resulting in a complete model (up to the threshold $t_q$). We chose a box for this example because it is an easy object to grasp and is unlikely to fall over.

To see further information about this project including example 3D models and videos, visit our webpage at
http://www.cs.uw.edu/robotics/3d-in-hand



(a) Box after one grasp. (b) Candidate grasps from by OpenRAVE, in descending order by view quality (top left to bottom right).

Fig. 6. Regrasp planning for a cracker box. Grasps having less overlap with previously occluded regions tend to score higher.

## VI. CONCLUSIONS AND FUTURE WORK

We developed a system that enables a robot to grasp an object and to move it in front of its depth camera so as to build a complete surface model of the object. To guide manipulator motion, we derived an information gain based variant of the next best view algorithm. By maximizing information gain, the manipulator moves the object so that the most uncertain surface areas come into the view of the depth camera. By incorporating occlusions caused by the manipulator, our technique can also determine when and how the robot has to re-grasp the object in order to build a complete model.

The experiments demonstrate that our approach can guide a manipulator such that it generates object models that are as complete as possible, given manipulator occlusions. Once the information gain falls below a threshold, the robot places the object back on the table and starts a grasp planner to generate possible grasp samples for the partially complete model. We show that our information based approach can naturally generate a ranking of the proposed grasps so as to minimally occlude still uncertain object surfaces.

Our current system has several limitations. Thus far, we have relied on external grasping techniques. In future work, we intend to learn better grasp strategies from experience, where a robot repeatedly tries to grasp objects and, once it succeeds, builds 3D models of them. These 3D models can then be used as repositories and training data for later grasping. Another interesting direction for future work is to use the modeled objects for object recognition and detection, thereby enabling a robot to learn novel objects and then find and effectively manipulate them once they are seen again in an environment.

## REFERENCES


[1] A. Saxena, J. Driemeyer, and A. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research (IJRR)*, vol. 27, no. 2, 2008.

[2] A. Collet Romea, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2009.

[3] D. Berenson and S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Proc. of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2008.

[4] B. Rasolzadeh, M. Bjorkman, K. Huebner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in real world," in *International Journal of Robotics Research (IJRR)*, 2009.

[5] J. Glover, D. Rus, and N. Roy, "Probabilistic models of object geometry with application to grasping," in *International Journal of Robotics Research (IJRR)*, vol. 28, 2009, pp. 999–1019.

[6] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in hand 3d object modeling," University of Washington, Tech. Rep. UW-CSE-10-09-01, 2010. [Online]. Available: http://www.cs.washington.edu/robotics/3d-in-hand

[7] PrimeSense, "http://www.primesense.com/."

[8] Microsoft, "http://www.xbox.com/en-US/kinect."

[9] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Comput. Surv.*, vol. 35, no. 1, pp. 64–96, 2003.

[10] C. Connolly, "The determination of next best views," in *Robotics and Autonomous Systems*, vol. 2, March 1985, pp. 432–435.

[11] N. Massios and R. Fisher, "A best next view selection algorithm incorporating a quality criterion," in *British Machine Vision Conference*, 1998.

[12] R. Triebel, B. Frank, J. Meyer, and W. Burgard, "First steps towards a robotic system for flexible volumetric mapping of indoor environments," in *Proc. of the 5th IFAC Symp. on Intelligent Autonomous Vehicles*, 2004.

[13] T. Foissotte, O. Stasse, A. Escande, P.-B. Wieber, and A. Kheddar, "A two-steps next-best-view algorithm for autonomous 3d object modeling by a humanoid robot," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1078–1083.

[14] H. Zha, K. Morooka, and T. Hasegawa, "Next best viewpoint (nbv) planning for active object modeling based on a learning-by-showing approach," *Lecture Notes in Computer Science*, vol. 1352, pp. 185–192, 1997.

[15] M. K. Reed, P. K. Allen, and I. Stamos, "Automated model acquisition from range images with view planning," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society Press, 1997, pp. 72–77.

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

[17] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 1996.

[18] B. Curless, "New methods for surface reconstruction from range images," Ph.D. dissertation, Stanford University, 1997.

[19] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," Robotics Institute, Tech. Rep. CMU-RI-TR-08-34, July 2008. [Online]. Available: http://openrave.programmingvision.com

[20] R. Balasubramanian, L. Xu, P. Brook, J. Smith, and Y. Matsuoka, "Human-guided grasp measures improve grasp robustness on physical robot," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, May 2010, pp. 2294–2301.