Exploiting Segmentation for Robust 3D Object Matching

Michael Krainin Kurt Konolige Dieter Fox

***Abstract*— While Iterative Closest Point (ICP) algorithms have been successful at aligning 3D point clouds, they do not take into account constraints arising from sensor viewpoints. More recent beam-based models take into account sensor noise and viewpoint, but problems still remain. In particular, good optimization strategies are still lacking for the beam-based model. In situations of occlusion and clutter, both beam-based and ICP approaches can fail to find good solutions. In this paper, we present both an optimization method for beam-based models and a novel framework for modeling observation dependencies in beam-based models using over-segmentations. This technique enables reasoning about object extents and works well in heavy clutter. We also make available a ground-truth 3D dataset for testing algorithms in this area.**

I. INTRODUCTION

The problem of aligning 3D models to scenes is a common one in robotics, with typical applications being tabletop manipulation [1], [2], object tracking [3], and articulated pose estimation [4], [5]. The problem is usually broken up into two phases. In the first, the detection phase, the presence of an object or part is inferred and one or more hypothesized (rough) poses are generated. In the second, the pose estimation phase, more precise poses are produced after exploring the space of poses and comparing candidates. In this paper, we examine the problem of pose estimation, with a particular focus on heavy clutter and occlusion. We assume a sensor that produces dense range images, such as the Microsoft Kinect.

The standard method for reasoning about alignment of 3D models in range data is Iterative Closest Point (ICP). Since ICP looks only at 3D points, it throws away information about the viewpoint from which the scene points were generated. As an alternative, beam-based sensor models can be used to take into account the protrusion of the model into free-space in the scene (e.g., left side of Fig. 1). While beam-based models can more precisely align object models by reasoning about free-space and occlusion, they suffer from several drawbacks, especially in cluttered scenes. In this paper, we present a practical method for pose alignment in the presence of clutter and occlusion that extends existing beam models by enforcing consist reasoning among dependent observations. We provide the following contributions:

- We propose a novel formulation of beam-based probabilistic sensor models which eliminates many of the false optima which occur in existing models. We do this by relaxing the standard beam independence assumption and exploiting the regularity of environments that is reflected in range data. This technique allows us to reason about extents of physical objects in a sensor model framework.
- We present a gradient-based search method for pose optimization using beam-based sensor models. We show that this technique is more consistently able to produce high-quality poses than ICP in the presence of clutter and occlusion.
- The code from this paper is available as an open source ROS[1] package with Ecto[2] Python bindings. We also make available all test data used in our evaluations. Unlike existing datasets for pose estimation from range data, ours emphasizes heavy clutter and occlusion.

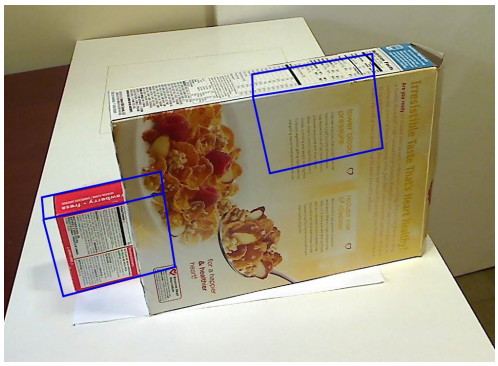


Fig. 1: Scene with clutter and occlusion. Matching the small box based on its 3D model is problematic. ICP will match the model with free-space protrusions, as on the occluded box at left. The beam model will find the correct match there, but will prefer to embed the smaller box model into the larger box, where the whole front surface matches.

We begin in Section II with an overview of related work. We then review beam-based sensor models in Section III and present a gradient-based approach to optimizing these models. In Section IV we introduce our segmentation-based sensor model to overcome some of the short-comings of independent beam models. Section V contains experimental results. Conclusions and future work are discussed in Section VI.

II. RELATED WORK

A broad range of techniques exist for performing object pose estimation. These vary both in the error function and

M. Krainin and D. Fox are with the Department of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA. {mkrainin,fox}@cs.washington.edu

K. Konolige is with Willow Garage Inc., Menlo Park, CA 94025, USA. konolige@willowgarage.com

This work was funded in part by ONR MURI grant N00014-09-1-1052.

[1] http://ros.org

[2] http://ecto.willowgarage.com

the search algorithm used to optimize it.

Some examples of criteria used in error functions are 2D point features [1], 3D point features [6], [7], and Chamfer matching [8]. Point feature matching, typically combined with RANSAC for geometric consistency, give good results provided sufficiently many and sufficiently distinctive features can be detected. Chamfer matching and related techniques can provide reliable detection even under occlusion but tend to have difficulty estimating full 6 DoF poses without multiple viewpoints.

Most commonly used with range data are error functions for explicit matching of surface geometry. The Iterative Closest Point (ICP) algorithm [9] attempts to minimize the sum of squared distances of points from the model to their respective closest points in the range image. ICP therefore makes use of range image pixels which have been given correspondences to the model but ignores others such as measurements of the background that provide information about the free-space in the scene.

Another class of error function for pose estimation is beam-based probabilistic sensor models (e.g., [10], [11], [12]). Unlike the ICP error metric, beam-based sensor models explicitly treat cases such as occlusion and measurements beyond the expected surface. We show that this metric actually performs quite poorly at selecting among distinct local optima, especially in the presence of occlusion. We propose a relaxation to the beam independence assumption used in these sensor models that removes false optima such as shown in the right-hand side of Fig. 1.

To our knowledge, no one has previously used gradient-based search over beam-based sensor models, likely because they are notorious for their discontinuous nature. Existing techniques are purely sample-based, relying for instance on coarse-to-fine grid search a single dimension at a time [11] or annealed particle filters [3]. Despite all the caution regarding the use of gradient-based search, we have found that with only minimal considerations for smoothness, gradient-based search over a beam-based sensor model produces correct poses more consistently than ICP. The more difficult problem appears to be in selecting among a handful of local optima, for which we propose a novel sensor model, relaxing standard beam-independence assumptions.

III. Independent Beam Model

We begin by reviewing beam-based sensor models and then present a gradient-based approach for optimizing over these models.

A. Beam Model

Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be measurements from a single frame of our depth sensor. For a given model $\mathcal{M}$, our goal is to find the transformation T^* of the model which best aligns the model with the sensor data. Because the sensor data comes from a viewpoint via a set of sensor beams, it is called a beam model.

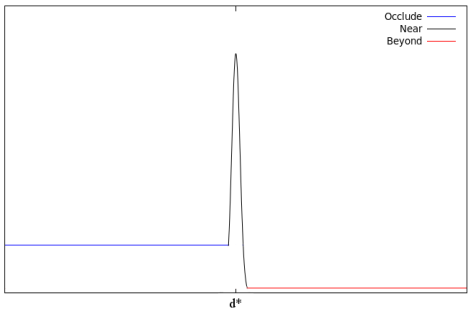


Fig. 2: Piecewise sensor model used in IBM. On the x-axis is the beam measurement d_i. d_i^* is the expected measurement from the model. Not to scale.

We formulate the problem in a maximum likelihood framework, in which T^* is estimated as

$$T^* = \arg\max_T p(\mathcal{D}|\mathcal{M}, T). \tag{1}$$

The standard assumption of mutually independent beams, which we will refer to as the Independent Beam Model (IBM), yields

$$p(\mathcal{D}|\mathcal{M}, T) = \prod_i p(d_i|\mathcal{M}, T). \tag{2}$$

The term $p(d_i|\mathcal{M}, T)$ can in principle consider many types of information, for instance depths, normals, and colors (see [13], [12]). For this paper, we just consider the depth component. Given $\mathcal{M}$ (e.g., a triangle mesh) and T, we render a mesh model of the object, resulting in a virtual depth map. Each measurement d_i has a corresponding expected measurement d_i^* in the virtual depth map.

When the expected measurement $d_i^* \neq \emptyset$, we use the 3-component, piecewise model shown in Fig. 2. This function is uniform for $d_i \ll d_i^*$ (occlusion of the expected surface), Gaussian about d_i^* with standard deviation σ_d for $d_i \approx d_i^*$ (near the expected surface) and uniform low probability for $d_i \gg d_i^*$ (beyond the expected surface). The crossover between components occurs where the Gaussian and uniform components are equal.

For $d_i^* = \emptyset$, we use a uniform distribution out to a maximum range. There is also some probability assigned to invalid sensor measurements. Beam models such as this are commonly used in the robotics literature [10].

Using IBM for pose estimation has two nice properties:

1) It handles occlusions by allowing beams to terminate in front of the model without too large a penalty. Thus the large value of the distribution in front of d_i^*.
2) It handles protrusions into free-space by penalizing beams that go through the model with a larger penalty. Hence in Figure 1, the match on the left-hand side would be penalized relative to the correct fit.

B. Optimization

Optimization involves finding the pose that maximizes the likelihood in Equation 1. Converting this to negative log

likelihoods from Equation 2 yields the sum

$$T^* = \arg\min_T \sum_i -log[p(d_i|\mathcal{M}, T)]. \quad (3)$$

If p were a Gaussian distribution, the above equation could be rewritten as minimizing a quadratic cost term (the exponent of the Gaussian), and handed to a standard nonlinear least-squares solver. Even in the case of a non-Gaussian PDF, the maximum likelihood result can be estimated by using the negative log likelihood as the cost term [14]. The advantage of doing this is that we can leverage existing algorithms, thereby simplifying implementation and improving extensibility. Nonlinear least squares requires Jacobians, which we compute numerically by reprojecting the model with slight changes in its transform. Surprisingly, there seem to be no examples of using nonlinear least squares optimization with the beam model in the literature.

Though continuous in d_i and in d_i^*, the independent beam model is discontinuous with respect to motions of $\mathcal{M}$ which cause d_i^* to jump from a finite value to $\emptyset$ or vice versa. Ganapathi et al. [11] take the approach of evaluating each beam over a small pixel window, taking $\max_j[\log p(d_i|d_j^*(\mathcal{M}, T)) + \lambda(i, j)]$, where λ penalizes the selection of a different pixel than i. We found we achieved better results by instead summing over the window as

$$p(d_i|\mathcal{M}, T) = \sum_{j \in \mathcal{W}} p(i, j) \cdot p(d_i|d_j^*(\mathcal{M}, T)), \quad (4)$$

where $p(i, j)$ is a 2D isotropic Gaussian with standard deviation σ_p and $\mathcal{W}$ is a small pixel window; we have found 3x3 to be sufficient. While the function is still discontinuous with respect to the motions described above, the windowing introduces gradations which we have found to improve the optimization's convergence.

For nonlinear optimization, we utilize the g^{2}o graph optimization framework [15], essentially a Levenberg-Marquardt technique. The graph contains vertices for the sensor and the object model poses. Each pixel in the image is represented as an edge connecting the two vertices. Associated with each edge is the negative log likelihood for that pixel given $\mathcal{M}$ and T. Error function and (numeric) Jacobian evaluations during the optimization prompt re-rendering of $\mathcal{M}$ in new poses, followed by computations of negative log likelihoods. By using the g^{2}o framework, it becomes quite straightforward to later add additional types of constraints such as feature correspondences if desired.

The optimization technique works well vis-a-vis ICP, and we compare them in Section V-B.1. It is usually able to find the correct pose for the model, given a strategy for recovering a set of local minima around the initial pose estimate. An example of the improvement given by IBM optimization is shown in Fig. 3.

IV. Segmented Beam Model

While IBM performs well in finding good local minima, it has problems in evaluating those minima to find the best one. As an example, consider the scenario shown in Fig. 1. Here

Fig. 3: Example usage of free-space constraints. (left) Box against which to align. (center) Incorrect pose resulting from ambiguity in the ICP error function. (right) Free-space constraints in the IBM optimization resolve the ambiguity.

IBM selects the best pose as one that embeds the smaller box in the larger one, since a larger percentage of the beams actually match the model, rather than being penalized for occlusion as in the correct match. One way to understand this problem is to say that scene surfaces matching the model should not extend beyond the model. The individual beams are not independent, but are considered as belonging to a coherent set of surfaces.

To formalize this idea, we use an (over-)segmentation of $\mathcal{D}$ into locally consistent segments $\mathcal{S} = \{S_1, \ldots, S_M\}$ to provide a better approximation to the likelihood function. The segments in $\mathcal{S}$ should be mutually exclusive, collectively exhaustive, and sufficiently fine-grained to ensure that measurements belonging to distinct objects in the environment will not combine into a larger segment. We will describe one technique to achieve such a segmentation in Section V. Given a segmentation $\mathcal{S}$, we approximate the data likelihood as

$$p(\mathcal{D}|\mathcal{M}, T) = \prod_{S_i \in \mathcal{S}} p(S_i|\mathcal{M}, T). \quad (5)$$

Let m_i be an indicator variable for whether sensor readings within the segment S_i were generated from the model $\mathcal{M}$. We compute the segment likelihood $p(S_i|\mathcal{M}, T)$ by marginalizing over m_i:

$$p(S_i|\mathcal{M}, T) = \sum_{m \in \{0,1\}} p(S_i, m_i = m|\mathcal{M}, T) \quad (6)$$

$$= \sum_{m \in \{0,1\}} p(S_i|\mathcal{M}, T, m_i = m) \cdot p(m_i = m|\mathcal{M}, T). \quad (7)$$

$p(m_i|\mathcal{M}, T)$ is a prior probability that a segment is generated from the model rather than, for instance, background or occluders in the scene. The performance of our algorithm is quite insensitive to the value of this term; for simplicity, we set it to 0.5.

$p(S_i|\mathcal{M}, T, m_i)$ is the segment likelihood conditioned on whether the segment was generated by $\mathcal{M}$. By performing this segment classification, we can evaluate the entire segment according to either a peaked distribution (for $m_i = 1$) or a broad distribution (for $m_i = 0$). The typical independence assumption of (2) would instead force us to make this decision independently for each pixel, resulting in problem cases such as in Fig. 1.

Only after the intermediate step of (7) do we proceed to apply a mutual independence assumption to the individual

beams. That is, the beams are considered conditionally independent given m_i:

$$p(S_i|\mathcal{M}, T, m_i) = \prod_{d_j \in S_i} p(d_j|\mathcal{M}, T, m_i). \quad (8)$$

As before, $p(d_j|\mathcal{M}, T, m_i)$ is the probability for an individual beam, now given a segment interpretation. If $m_i = 1$, we expect the measurement to have been generated by $\mathcal{M}$ and apply a Gaussian sensor model over the depth. The negative log likelihood (up to a constant) is then $\min((d_j - d_j^*)^2/(2\sigma_d^2), t)$, where t is a cutoff for robustness. If $m_i = 0$, we instead use uniform distributions corresponding to those used in IBM (the blue and red lines in Fig. 2).

The Segmented Beam Model (SBM) penalizes embedding a smaller model surface within a larger scene surface. Figure 4 shows a typical example where SBM eliminates a false positive produced by IBM.

The careful reader may ask why we did not directly optimize SBM. As SBM includes an assignment step in the form of the indicator variables, a more sophisticated optimization technique like expectation maximization may be called for. We leave the investigation of this problem as future work.

V. RESULTS

We implemented the optimization algorithm in C++ with model rendering using OpenGL and per-pixel negative log likelihoods evaluated using CUDA. As currently implemented, it takes approximately 1 millisecond to render the depth map and subsequently compute the per-pixel negative log likelihood map on an NVIDIA GeForce GTS 450 graphics card. We perform approximately 500 such renderings per nonlinear optimization. Other systems for evaluating sensor models on graphics cards (e.g. [11]) suggest that additional optimizations and a high-end graphics card could give at least an order of magnitude improvement in speed.

Evaluating SBM requires a segmentation $\mathcal{S}$. We implemented a simple raster-scanning connected components algorithm, with thresholds for out-of-plane distance (1.5 mm) and difference in normals (8°). We compare with neighbors 2 pixels away rather than the immediate neighbor since the normal estimation procedure has a strong smoothing effect. The thresholds are set such that under-segmentation is rare. A downside of the connected components approach is that small "bridges" can connect otherwise quite distinct segments, so the thresholds must be set conservatively. More sophisticated segmentation algorithms may reduce the need to over-segment. Over-segmentation is acceptable; in the limit of single pixel segments, SBM reverts to IBM.

A. Qualitative Properties of the Sensor Models

We examine the difference between IBM and SBM in terms of their response in cluttered scenes. Fig. 4 depicts a small juice carton and a larger cereal on a table (top left). For IBM, there are two local minima for the juice carton: one on the juice carton, and one embedded in the larger cereal box. The middle image is a response map along XY dimensions, as the juice carton model is moved around the table. Note the very broad line where the juice carton is embedded in different parts of the cereal box. IBM does not penalize model surfaces that are embedded in larger surfaces, and so the response is the same as where the juice carton matches to itself. In the presence of clutter, this model tends to generate false positives for pose estimates on other surfaces, especially if the correct match is partially occluded.

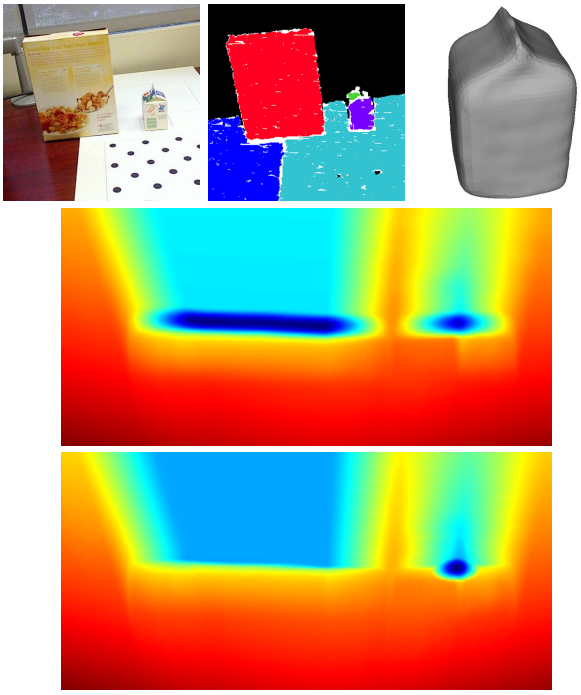


Fig. 4: Matching a juice carton into a scene. (top left) Frame being matched against. (top center) Segmentation of the range image. (top right) Carton mesh model. (middle, bottom) Error functions for the Independent Beam Model and the Segmented Beam Model respectively. Heat maps depict negative log likelihoods as the carton model is translated along the two degrees of freedom of the table plane; red is high cost, blue is low cost. Here, we restrict to two degrees of freedom for visualization purposed only. Notice that SBM eliminates the incorrect minimum along the face of the box.

By contrast, the segmentation model SBM, shown in Fig. 4 bottom, has a clear minimum on the juice carton. The elongated minimum is completely eliminated, since embedding the juice carton model in the larger cereal box segment is penalized by (7). Note that the response map is very similar in other respects.

B. Pose Estimation

Few datasets currently exist for matching 3D models into range images, especially with clutter and occlusion. The RGB-D Object Dataset [16] contains some cluttered scenes; however, it is labeled for object detection, not for pose estimation. The Solutions In Perception Challenge[3] is labeled for pose estimation but includes only very minimal

[3] http://opencv.willowgarage.com/wiki/SolutionsInPerceptionChallenge

Abbreviation	Object
A	All detergent
Cl	Clorox bleach
Co	Coke can
OJ	OJ carton
So	Soup can
Sp	Spam can
Ti	Tilex spray
To	Toothpaste bottle
Z	Ziploc bags

TABLE I: The nine objects included in our test data along with their abbreviations used in later figures.

clutter and occlusion. We therefore elected to collect our own test data[4]. For each of nine objects listed in Table I, we collected an average of five scenes. A scene consists of a range image of an object in the presence of clutter and/or occlusion. Additionally, each scene contains an RGB image and a ground truth object pose derived from a calibration pattern in the image.

In the following experiments, we perform multiple trials per scene, with each trial having a different initial pose. The initial poses are generated by perturbing the ground truth pose by 2 to 4 centimeters in translation and 20 to 30 degrees in rotation about a randomly selected axis. For both ICP and our IBM optimization, we allow 20 random restarts. We deem a trial to be successful if the resulting pose is within 1.5 centimeters in translation and 10 degrees in rotation. Rotationally symmetric objects are not penalized for rotations about their axis of symmetry. The results in Table II and Table III are generated using a Gaussian approximation to the posterior beta distribution over frequency correct.

1) Optimization Results: Before considering how to distinguish between local optima, we need to make sure we are generating the correct optimum as one of the candidates. In our first pose estimation experiment, we compare ICP alignment to the IBM gradient-based optimization described in Section III-B. The ICP algorithm we use includes a number of common tweaks including a point-to-plane error metric, hard thresholding on correspondence length, rejection of boundary-point correspondences, and trimming of the longer remaining correspondences. The ICP optimization was also performed using g^{2}o's LM implementation.

In Table II, we estimate the frequency with which the two optimization techniques include a correct pose among their 20 restart results; the overall frequency of correct poses is bounded by this value. Across the board, IBM does at least as well and in many cases better than ICP.

We show some example failure cases in Fig. 5. In ICP, problems can arise such as inability to resolve a particular degree of freedom (as in Fig. 5a), or being pulled away by a nearby surface. IBM avoids many of these problems by penalizing poses which place parts of the model into volumes that have been observed as unoccupied. Fig. 5b shows an example where these cues are largely unavailable due to the occlusion at the ends of the Ziploc box. As a result, the model is improperly translated so that one face matches the surface of the cereal box. Cases like this, where a model can span multiple object surfaces without violating free-space constraints, suggest there might be some value in adding segmentation information into the local optimization in addition to the global evaluation.

[4]All code and test data for this paper is available at: http://ros.org/wiki/Papers/ICRA2012_Krainin_Konolige_Fox

(a) ICP

(b) IBM

Fig. 5: Example failure cases for the two optimization techniques; original scene on the left, pose optimum on the right, with the Ziploc box model overlaid in red. In ICP (top), free-space protrusions occur because free-space information is not used. IBM failures (bottom) can occur when occlusion is present blocking all edges along one direction.

2) Evaluation Results: In Table III, we present the frequency of correct poses for different evaluation functions. Each function is given as input the random restart results from the IBM optimization. In almost all cases, SBM performs better or at least as well as the other evaluation functions. As illustrated in Fig. 6a, both ICP Mean Squared Error (ICP MSE) and IBM can match to similarly shaped objects in cluttered scenes. This is particularly a problem for IBM because when the true surface is occluded, a similarly shaped but unoccluded surface results in a higher likelihood.

SBM solves this problem by eliminating most alternative optima based on surface size criteria. Of course, if other, similarly sized and similarly shaped surfaces exist in the scene, the corresponding optima may still be selected. SBM does introduce the possibility for a different type of failure caused by under-segmentation. An example is shown in Fig. 6b in which there is insufficient difference in normals or depth to distinguish the Ziploc box from the cereal box. The result is that matches using this surface are penalized, and an erroneous pose is selected instead.

Table III clearly demonstrates the advantage of using SBM over IBM, but it is important to note that IBM is simply a special case of SBM (Fig. 7c). Segmentations can vary from being very over-segmented to being under-segmented, and

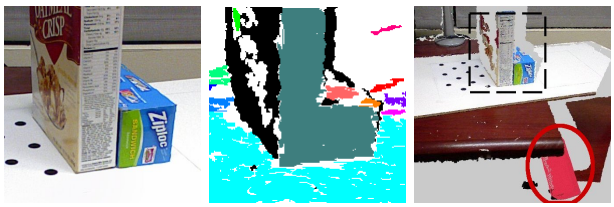
Optimization	A	Cl	Co	OJ	So	Sp	Ti	To	Z	Total
ICP	0.85 ± 0.10	0.98 ± 0.04	0.81 ± 0.11	0.73 ± 0.11	0.87 ± 0.09	0.88 ± 0.09	0.98 ± 0.05	0.83 ± 0.10	0.79 ± 0.10	0.86 ± 0.03
IBM	$\mathbf{0.88} \pm 0.09$	0.98 ± 0.04	$\mathbf{0.98} \pm 0.04$	$\mathbf{0.84} \pm 0.09$	$\mathbf{0.98} \pm 0.04$	$\mathbf{0.94} \pm 0.06$	0.98 ± 0.05	$\mathbf{0.85} \pm 0.10$	$\mathbf{0.82} \pm 0.09$	$\mathbf{0.93} \pm 0.02$

TABLE II: Frequency of correct pose estimates being among the random restart results. 95% confidence intervals.

Evaluation	A	Cl	Co	OJ	So	Sp	Ti	To	Z	Total
ICP MSE	$\mathbf{0.88} \pm 0.07$	0.99 ± 0.03	0.65 ± 0.11	0.66 ± 0.10	0.92 ± 0.06	0.70 ± 0.10	0.98 ± 0.03	0.62 ± 0.11	$\mathbf{0.74} \pm 0.09$	0.79 ± 0.03
IBM	0.57 ± 0.11	0.99 ± 0.03	0.66 ± 0.10	0.51 ± 0.10	0.75 ± 0.10	0.60 ± 0.11	0.98 ± 0.03	0.48 ± 0.11	0.54 ± 0.10	0.67 ± 0.04
SBM	0.74 ± 0.10	0.99 ± 0.03	$\mathbf{0.96} \pm 0.04$	$\mathbf{0.75} \pm 0.09$	$\mathbf{0.99} \pm 0.03$	$\mathbf{0.95} \pm 0.05$	0.98 ± 0.03	$\mathbf{0.81} \pm 0.09$	0.55 ± 0.10	$\mathbf{0.85} \pm 0.03$

TABLE III: Frequency of correct pose estimates for various evaluation techniques for selecting between random restart-based proposals. 95% confidence intervals.

(a) ICP MSE + IBM



(b) SBM

Fig. 6: Example failure cases for the restart evaluation functions. (a) ICP MSE and IBM sometimes select matches to similarly shaped, though differently sized objects, as with this toothpaste bottle. (b) Under-segmentation in SBM leads to a lack of surfaces against which to match. Here the Ziploc bags and cereal box are segmented as the same object.

this has an effect on the quality of the resulting sensor model. Fig. 7a shows the frequency of correct poses on the Spam can data as a function of the granularity of our segmentation. We vary the angular threshold of our connected components algorithm to produce a range of segmentations (examples in Fig. 7c-e). As this figure demonstrates, we are able to achieve substantial improvement over IBM without requiring a perfect segmentation. As the threshold continues to increase, we eventually see some under-segmentation (Fig. 7e).

C. Object Model Selection

Finally, we consider the problem of distinguishing the identity of an object based on the pose quality metrics we have presented. In applications such as grasping, the problem may arise that an object is known to be in a certain region, but its identity is unknown [17] (see Fig. 8).

In the experiment shown in Table IV, we aligned each of the nine object models to each of the scenes using the IBM optimization. We then scored the poses of each model using one of the three evaluation functions to select a model. Table IV presents the confusion matrices for the three evaluation functions.

IBM performs much better than ICP MSE in terms of object confusion even though ICP MSE proved the better of the two techniques for pose selection in Table III. In the context of model selection, IBM's behavior of increasing its score with greater numbers of near-surface beams is beneficial rather than detrimental. Whereas ICP MSE suffers from matching a small object (the toothpaste bottle) to larger surfaces, IBM selects the appropriately sized object because it explains more of the beams. So in this problem, IBM has many of the desirable properties as SBM, and the two perform fairly comparably.

VI. CONCLUSIONS AND FUTURE WORK

We have presented an extension to standard beam-based models which, through the addition of another layer to the probabilistic model, incorporates segment information into its likelihood function. The key insight is that we can exploit the regularities in our sensor data to consistently reason about pixels both inside and outside of the model silhouette. Our model provides a notion of surface extents, which we have shown has a major impact on the ability to correctly select between optima in cluttered and occluded scenes.

We have also presented a gradient-based approach to optimizing a beam-based sensor model. We used this technique to generate candidates for our Segmented Beam Model and showed its advantages over ICP for these purposes. Finally, we constructed a pose estimation dataset focusing on scenes with heavy clutter and occlusion. We have made available all code and test data associated with this project.

In the future, there are a number of interesting directions to explore. We hope to examine the advantages of different segmentation algorithms to help avoid problematic under-segmentations. Though the problem of generating perfect segmentations is far from solved, we believe that consistently over-segmenting is a much simpler problem. Also beneficial would be to detect failed segmentations based on the number of pixels marked as being generated by the model; this would trigger a more conservative segmentation.

A number of extensions such as color information could be used in conjunction with SBM. Color images could be applied both to improve segmentation and to better discriminate between local optima. It is worth noting that the objects in our test data are all highly textured (they are the same objects used in the Solutions In Perception Challenge), so color information would likely give an unusually large performance gain.

	A	Cl	Co	OJ	So	Sp	Ti	To	Z
A	2		1					2	
Cl		2						3	
Co								5	
OJ				5		1			
So					2	1		2	
Sp						5			
Ti				1			1	1	1
To								5	
Z									6

(a) ICP MSE

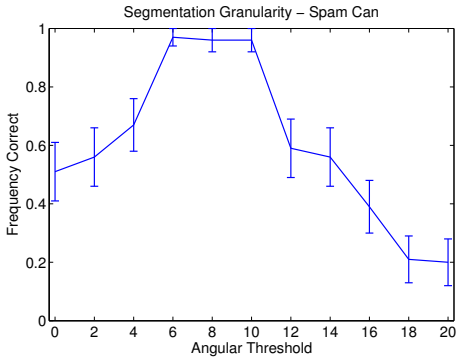
	A	Cl	Co	OJ	So	Sp	Ti	To	Z
A	5								
Cl		5							
Co			2			1		2	
OJ				5					1
So					4			1	
Sp						4			1
Ti							4		
To								5	
Z						1			5

(b) IBM

	A	Cl	Co	OJ	So	Sp	Ti	To	Z
A	4							1	
Cl		5							
Co			4					1	
OJ				6					
So					5				
Sp						5			
Ti							4		
To								5	
Z				1		1			4

(c) SBM

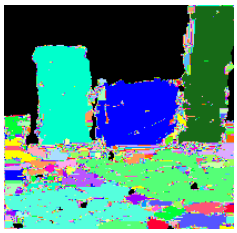
TABLE IV: Confusion matrices for nine objects. Model poses from the IBM optimization were evaluated using the three evaluation functions. Total correct of 28, 39, and 42 respectively (out of 46).



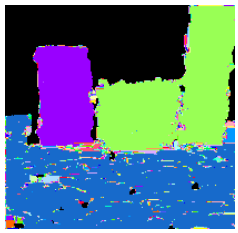
(a)

(b)

(c)



(d)



(e)

Fig. 7: (a) Frequency of correct pose estimates as a function of the granularity of the segmentation. Finer segmentation occurs for smaller values of the angular threshold. (b) Example frame from the dataset. (c)-(e) Segmentations for thresholds of 0°, 8°, and 12° respectively. Note the under-segmentation in (e).

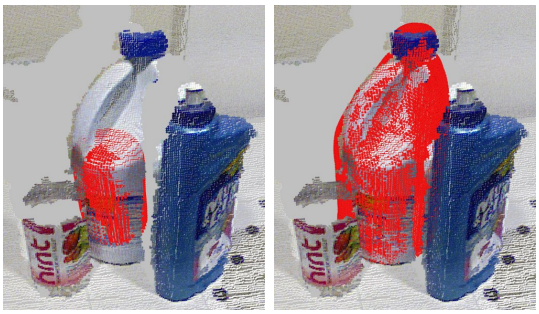


Fig. 8: Example object identity determination from Table IV. The question is whether the data is better explained by a Coke can (left) or a Clorox bottle (right).

REFERENCES

[1] A. Collet Romea, M. Martinez Torres, and S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *Int'l Journal of Robotics Research (IJRR)*, April 2011.

[2] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2011.

[3] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots, Special Issue: Selected papers from Robotics: Science and Systems 2008*, vol. 26, no. 2–3, pp. 123–139, April 2009.

[4] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3D human body tracking with an articulated 3D body model," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2006.

[5] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *Int'l Journal of Robotics Research (IJRR)*, vol. 30, no. 11, pp. 1311–1327, September 2011.

[6] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 5, 1999.

[7] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. of the IEEE/RSJ Int'l Conference on Intelligent Robots and Systems (IROS)*, 2010.

[8] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, A. Agrawal, and H. Okuda, "Pose estimation in heavy clutter using a multi-flash camera," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2010.

[9] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, pp. 239–256, February 1992.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, September 2005, iSBN 0-262-20162-3.

[11] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] E. Herbst, P. Henry, X. Ren, and D. Fox, "Toward object discovery and modeling via 3-D scene comparison," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2011.

[13] Y. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, "Multi-view image and tof sensor fusion for dense 3-D reconstruction," in *Proc. of the Int'l Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2009.

[14] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzibbon, "Bundle adjustment - a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS. Springer Verlag, 2000, pp. 298–375.

[15] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2011.

[16] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. of the IEEE Int'l Conference on Robotics & Automation (ICRA)*, 2011.

[17] K. Hsiao, M. Ciocarlie, and P. Brook, "Bayesian grasp planning," in *ICRA 2011 Workshop on Mobile Manipulation*, 2011.