Generalized 3D Object Representation using Bayesian Eigenobjects

Benjamin Burchfiel and George Konidaris
Duke University, Department of Computer Science, {bcburch, gdk}@cs.duke.edu

***Abstract*—Current methods for representing 3D objects for robotic interaction have significant limitations. They do not allow knowledge transfer from previously encountered objects to similar but novel objects, they construct large object databases that do not scale and are expensive to query, or they require hand tuned object models. We propose the use of Variational Bayesian Principal Component Analysis (VBPCA) directly on 3D object representations to create compact low-dimensional probabilistic models for classes of 3D objects. We show that these learned Bayesian Eigenobjects (BEOs) are well suited to important and practical robotic tasks including the classification and pose estimation of novel objects. Furthermore, we show that this approach can complete partially observed novel objects, allowing for the classification and pose estimation of novel occluded query objects.**

I. Introduction

It is inevitable that robots operating in the real world will be required to interact with previously encountered objects. While databases of object models exist with tens of thousands of objects, the world contains orders of magnitude more variation. Although 2D object detection has improved dramatically in the recent years, techniques for detecting and interacting with 3D objects are still quite limited. The common practice is still to build a library of 3D object models and match them to encountered objects in the world, often using ICP [18]. While this can be successful in highly controlled environments, it can not be feasibly scaled to less controlled settings where a wide variety of objects must be considered. Take, for example, a robot designed to clear dishes off of a table. The amount of variation in the size and shape of bowls, platters, and plates that the robot may encounter is huge. While such a task might be feasible for a specific set of place settings, creating a general purpose table clearer is beyond the current state of the art. Thus, new approaches are needed to allow robots to generalize across such highly variable objects.

There are several key building blocks of most robot applications involving interactions with objects in the world: object detection, pose estimation, and classification. These tasks form the perceptual foundation for many higher level operations including object manipulation and world-state estimation. In this work, we focus on the classification, pose estimation, and geometric completion of novel objects and demonstrate that Bayesian Eigenobjects (BEOs) are naturally suited to facilitating these tasks.

Our work uses Variational Bayesian Principal Component Analysis as the basis for a multi-class object representation. By learning a compact basis for each class, we are able to store previously encountered objects efficiently by storing only their projection coefficients. Furthermore, novel objects can be localized, classified, and completed by projecting them onto class basis and then projecting back into object space. Because we do not query a database of individual objects, our method scales gracefully with the number of objects in each class— requiring a constant amount of computation for projection and reconstruction even as the number of previously encountered data-points increases— and is able to handle objects of much higher resolution than competing methods.

A key advantage of our single, unified, object representation is its ability to perform partial object completion. Because objects in real environments are rarely observable in their entirety from a single vantage point, the ability to produce even a rough estimate of the hidden regions of a novel object can be extremely useful. Furthermore, being able to classify partial objects dramatically improves the efficiency of object-search tasks by not requiring the agent to examine all candidate objects from multiple viewpoints. Experimentally, we applied our method to several datasets consisting of scanned objects. We were able to successfully estimate the rotational pose of novel objects, reconstruct partially unobserved objects, and categorize novel objects by class. We also show that these completed objects are of suitable quality for use in classification and pose estimation.

II. Background

A. 3D Object Recognition

While significant work has been done on 3D object recognition of known objects [23, 19, 16, 10, 9, 14, 21], less progress has been made on representing classes of 3D objects in a general way. One class of related approaches are object-database approaches [25, 2, 24, 12, 22, 13]. These methods construct a large database of complete and high quality object scans. When a novel object is encountered, it is used as a query into the database. Commonly used features for matching include local point features [2], shape features [12], global features [25], or a mixture of local and global features [13]. While these types of approaches have appealing properties, they also have some limitations. Because the database explicitly contains high quality models of object instances, extremely accurate information on the query object is available if an exact match to the query object exists. This is very effective for tasks such as partially specified object completion [13]. A significant drawback exists, however, if an exact match is not found in the database. While some approaches still attempt

to find a nearest match in such a case [13, 2], the results will be poor if the query object is sufficiently different from any in the database. Looked at another way, instance-based database models are necessarily discrete, containing only a finite number of exemplars. If coverage of the object space is not sufficient (does not include enough objects) then it will yield poor results, in both match quality and in behavior transfer. Furthermore, because the database is explicitly composed of training examples, it necessarily scales with the size of the training input. On moderately sized datasets this is often not an insurmountable issue, but it can become a problem as both the size of the class model and query latency increase with the training size.

Another class of techniques consists of parts-based approaches. These methods learn a dictionary of parts and represent objects via a combination of parts [8, 20, 17]. A key advantage of parts-based approaches is compactness—a shared dictionary of common parts means that maintaining a database of all previously seen objects is unnecessary. Furthermore, by associating an attribute (such as an affordance) to parts, knowledge can be transferred to new objects. One drawback of parts-based approaches is their inability to reconstruct incomplete objects. Because objects are represented as a collection of parts, a partial object model will not generally specify what the hidden portion of the object geometry is. While this is not necessarily an issue for recognition tasks, it is a drawback in other contexts such as object interaction. Furthermore, complex models consisting of numerous parts may become computationally intractable to reason about as the number of possible configurations is exponential in the number of parts.

B. Variational Bayesian Principal Component Analysis

Our work uses Variational Bayesian Principal Component Analysis (VBPCA) to learn compact bases for classes of objects. VBPCA is an extension of probabilistic PCA (PPCA) [15], which models each datapoint as

$$\mathbf{x_i} = \mathbf{W}\mathbf{c_i} + \mu + \epsilon_i \quad \forall \mathbf{x_i} \in \mathbf{X}, \tag{1}$$

where $\epsilon_\mathbf{i}$ is zero mean Gaussian noise associated with datapoint i. PPCA also makes the assumption that each projected datapoint, $\mathbf{c_j}$, is generated from a zero mean Gaussian distribution. The model parameters for PPCA may be efficiently found using the EM algorithm, which alternates between updating the estimate of each datapoint's coefficient, $\mathbf{c_j}$, and updating W, μ, and ϵ. This probabilistic approach to PCA provides a density model which is well suited to density estimation and data compression. Bayesian PCA (BPCA) [5] further extends this model by introducing (Gaussian) priors (parametrized by $\mathcal{H}$) over the elements of μ and W. This allows BPCA to model the entire posterior probability of model parameters:

$$p(\mathbf{W}, \mu, \mathbf{C}|\mathbf{X}, \mathcal{H}). \tag{2}$$

Unfortunately, there is no analytic form for this probability, so straightforward application of the EM algorithm is problematic. VBPCA approximates this posterior probability as:

$$q(\mathbf{W}, \mu, \mathbf{C}) \approx p(\mathbf{W}, \mu, \mathbf{C}|\mathbf{X}, \mathcal{H}), \tag{3}$$

where $q(\mathbf{W}, \mu, \mathbf{C})$ is a factored approximation of the posterior: [4, 1]

$$q(\mathbf{W}, \mu, \mathbf{C}) = \prod_{i=1}^{d} q(\mu_\mathbf{i}) \prod_{i=1}^{d} q(\mathbf{w}_i) \prod_{i=1}^{n} q(\mathbf{c_i}). \tag{4}$$

This can be thought of as a regularized version of PPCA, providing the advantages of PPCA (including intrinsic density estimation) with increased resilience to over-fitting due to the prior. This property makes it especially well suited for situations where the dimensionality of the problem is high compared to the number of datapoints, i.e. $n <= d$, as is true in our case.

III. Method

Our approach is based on constructing a generative model for each class of objects, and then using that model to answer queries about novel or partial objects.

A. Class Models: Eigenobject Construction via VBPCA

We begin with a library of known objects of several classes, consisting of a complete 3D scan of each object. For each object in each class, we convert these scans into 3D voxel-based objects with a canonical orientation. We use coordinate descent congealing [11] to roughly align the objects in each class and then manually inspect and refine the alignment as needed. Congealing is a joint alignment algorithm that iteratively seeks to minimize a measure of group dissimilarity by warping group members at every iteration. We use element-wise binary entropy across all voxel objects in a class as our dissimilarity measure and allow rotation and translation transformations. Some of our data was sourced from ShapeNet [7] and generally arrived pre-aligned, while our manually scanned objects required alignment.

To construct a model for each class, we performed VBPCA on the registered 3D voxel representations of the class members using a manually specified basis size, k. Our hyperparameters were all zero mean, unit variance Gaussians which provided some regularization to the solution. A compact basis for the objects in a given class can be found by retaining only the first k rows of $\mathbf{W}$. Given a novel object in voxel-vector form, $\mathbf{o}$, its projection onto $\mathbf{W}$ can be found via

$$\mathbf{o}' = \mathbf{W}^T(\mathbf{o} - \mu). \tag{5}$$

Conversely, any point in the space of $\mathbf{W}$ can be converted back to a voxel object by solving

$$\hat{\mathbf{o}} = \mathbf{W}\mathbf{o}' + \mu. \tag{6}$$

We refer to $\hat{\mathbf{o}}$ as the "reconstructed" version of $\mathbf{o}$ and $\mathbf{o}'$ as the "projected" version of $\mathbf{o}$ (with respect to some class).

This formulation provides several benefits. First, we need not store or query an entire object database; instead, we need only store $\mathbf{W}$ and μ for each object-class. Second, we can

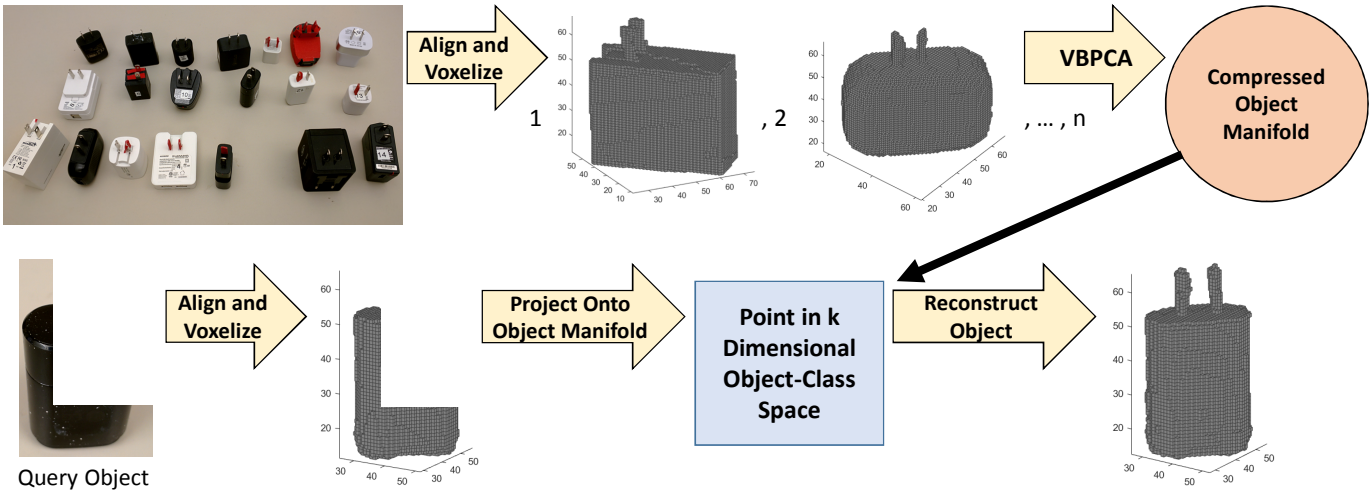

Fig. 1. The pipeline for obtaining a compressed representation of a query object. The upper left picture shows the objects used to create the plug-object manifold. The 10 most principal components and a mean vector were retained, leading to a 50 percent reduction in database size and capturing approximately 86 percent of variance in the input. The lower row shows an example query and reconstruction.

represent any object in a given class using a single coefficient vector of dimension k_i. In practice, $k \ll d$, providing an extremely compact representation.

Once a model is learned, it is not necessary to retain the original objects, $\mathbf{O}$, or their VBPCA coefficients, $\mathbf{O'}$. Instead, we retain an estimate of the learned (compressed) basis, $\mathbf{W}$, the estimated mean, μ, and the estimated variance of the noise, v_ϵ. Because it is assumed that the VBPCA coefficients for a given object, $\mathbf{o'_i}$ are normal, i.e.

$$p(\mathbf{o'_i}) = \mathcal{N}(\mathbf{o'_i} : 0, v_\epsilon(v_\epsilon \mathbf{I} + \mathbf{W}^T\mathbf{W})^{-1}), \tag{7}$$

this representation is also sufficient to provide a predictive density in the learned class manifold.

B. Pose Estimation

Pose estimation, the determination of an object's position and orientation, is necessary for many robotic manipulation and planning tasks. While it is relatively straightforward to acquire a rough estimate of object position given object detection and segmentation, determining orientation is more difficult. Using BEOs, it is possible to determine object orientation using a try-verify approach. We define a score based upon the L_2 error between two voxel objects,

$$r_i(\mathbf{o}, \hat{\mathbf{o}_i}) = 1 - \frac{||\mathbf{o} - \hat{\mathbf{o}_i}||_2}{|\mathbf{o}|}, \tag{8}$$

where a score of 1 denotes a perfect match and a score of 0 indicates that all voxels differ between the two objects. This score can be used to localize an object by projecting it, in various orientations, onto a basis and then picking the orientation that yields the highest reproduction score.

Let $\mathbf{O} = \mathbf{o_1}, \mathbf{o_2}, ...\mathbf{o_p}$ be object $\mathbf{o}$ in p orientations. To estimate the true orientation or o we solve

$$g(\mathbf{o}) = \arg\max_{\mathbf{o_i} \in \mathbf{o}} \quad r_i(\mathbf{o}, \hat{\mathbf{o}_i}). \tag{9}$$

In general, if a fine resolution orientation is required, there may be a large number of candidate poses. With three degrees of freedom for 3D rotations, discretizing to 1 degree of precision requires $46,656,000$ candidate poses. Fortunately, two mitigating factors make this less daunting. First, each query is totally independent and thus trivially parallelizable; it is possible to distribute the workload to multiple processors or accelerate it via GPU. Second, and more germane to this paper, searching all of $SE(3)$ is often unnecessary in practice. For instance, objects in an environment often have a canonical base upon which they sit. Leveraging this yields the common "up is up" assumption which reduces the space of possible orientations to those about the z axis only. Under such an assumption, 1 degree of pose precision only requires 360 candidate poses.

C. Object Classification

Another essential part of many robotic tasks is the classification of novel objects. Let the learned models for multiple classes be denoted $\theta_1, \theta_2, ..., \theta_\mathbf{m}$ where $\theta_\mathbf{i} = \{\mathbf{W^i}, \mathbf{mu^i}, v_\epsilon^i\}$ and let the novel query object be denoted $\mathbf{o_q}$. We wish to assign a label, l_i, to $\mathbf{o_q}$ from the set $L = \{1, 2, ..., m\}$ where l_i corresponds to class i. Intuitively, a key attribute of an object's true class is the error between the original object and the object obtained by projecting $\mathbf{o}$ into the basis learned for its true class and then projecting back into voxel space. Care must be taken in several areas, however, to create a successful 3D object classifier. Equation 8 works well for objects of the same class, but can be misleading when applied to objects of very different shape. Consider, for instance, the comparison of two significantly different objects, both with little solid volume. The first object is a bowl with thin sides while the second is a tall and thin flower. Because both objects have small filled volume, much of the unfilled space in the voxel representation will correspond between the two object models. Directly applying equation 8 in such a situation would lead to

misleadingly low error given how different the two objects are, making it unsuitable for comparing disparate objects across multiple classes.

To address this issue, we leverage a more nuanced representation of inter-object distance. Let $\mathbf{o}$ be the object to be classified and

$$\hat{\mathbf{o_i}} = \mathbf{W_i}\mathbf{W_i^T}(\mathbf{o} - \mu_\mathbf{i}) + \mu_\mathbf{i} \tag{10}$$

be the approximation obtained by projecting $\mathbf{o}$ onto the learned basis for class i and then projecting back into object space. From $\mathbf{o}$ and $\hat{\mathbf{o_i}}$ we extract the Euclidean Distance Transform [6] from each object, $\mathbf{D}$ and $\hat{\mathbf{D}}_\mathbf{i}$ respectively. Each distance transform forms a 3D matrix of the same dimensions as the object from which it was extracted. Each entry in the distance transform is the euclidean distance between its corresponding voxel in the original object and the closest filled voxel. As a result, entries that correspond to filled voxels have a value of 0 while entries that correspond to voxels far from filled portions of the object have high value. By computing the 2-norm of the difference between distance fields, we can create a more robust measure of distance between two objects. Using these distance norms, the distance reconstruction error between an object and its approximation when projected onto class i is

$$e_d(\mathbf{o}, \hat{\mathbf{o_i}}) = ||\mathbf{D} - \hat{\mathbf{D}}_\mathbf{i}||_2. \tag{11}$$

Because it implicitly captures shape differences between two objects, this distance error provides a much more robust metric for inter-class object comparison than equation 8.

Armed with this more versatile inter-object distance metric, we can now construct an object classifier. Let $\mathbf{O} = \mathbf{o^1}, \mathbf{o^2}, ..., \mathbf{o^N}$ be the set of all training objects used to find bases for all classes. We train a multi-class classifier over these objects with inputs $\mathbf{X}$ and $\mathbf{y}$ where element $\mathbf{x_{h,i}}$ corresponds to the reconstruction distance error obtained when projecting the hth object onto class i

$$e_d(\mathbf{o^h}, \hat{\mathbf{o_i^h}}) \tag{12}$$

and y_h is the true class label associated with the hth object. This classifier can be used to determine the class of a novel object, $\mathbf{o}$ by constructing a query vector,

$$\mathbf{x_o} = [e_d(\mathbf{o}, \hat{\mathbf{o_1}}), e_d(\mathbf{o}, \hat{\mathbf{o_2}}), ..., e_d(\mathbf{o}, \hat{\mathbf{o_m}})], \tag{13}$$

consisting of equation 11 applied to the query object and each of the m classes. Our implementation uses a multi-class linear-SVM [3], but any multi-class classifier suffices.

This formulation has several advantages over alternative classification strategies. First, unlike methods that operate directly on objects, there is no need to tune 3D features for individual classes. Furthermore, even objects of $250 \times 250 \times 250$ size contain over fifteen million voxels. Using such high dimensional input for classification is often infeasible, especially when the number of training examples is relatively small. Second, by leveraging reconstruction error instead of directly using projection coefficients, our approach eliminates the necessity of handling variable length classifier input. Unless each class has the same number of components, the projection coefficients obtained by projecting a single object onto multiple classes will not be the same length.

D. Partial Object Completion

In real-world environments, robots almost never have entire views of the objects they encounter. Even with the prevalence of multiple sensor, multiple modality, perception on modern robots, obtaining a complete 3D view of an encountered object requires sensing from multiple sides. If robots are to be mobile and operate outside of lab environments, it is unreasonable to expect that they will perceive objects from numerous vantage points before reasoning about them.

The alternative is to infer, from a partial model of an object and prior knowledge, what the remaining portions of the object may be. BEOs naturally provide this ability because they offer a generative representation of objects in a class. Since each learned basis provides an object-class manifold, if we are able to find the point on this manifold that best corresponds to a given partial object, reconstructing by projecting back into voxel object space will yield a prediction for the unobserved portions of the object.

Similar to Li et al. [13], we assume that the partial object, $\mathbf{o_p}$, consists of filled, empty, and unknown pixels. It is useful to define a n by k diagonal and binary selection matrix V such that $\mathbf{V}\mathbf{o_p} = \mathbf{w}$ where $\mathbf{w}$ is a length $k < d$ vector consisting of only the known (filled and empty) elements of $\mathbf{o_p}$. The ith element along the diagonal of $\mathbf{V}$ will be 1 if the ith element of $\mathbf{o_p}$ is known, and 0 otherwise. Let $\mathbf{o'_p}$ denote the smallest error projection of $\mathbf{o_p}$ onto the class basis $\mathbf{W}$. The error induced by an arbitrary projection $\mathbf{o'_i}$ with respect to $\mathbf{o_p}$ is

$$E(\mathbf{o'_i}) = ||V(\mathbf{W}\mathbf{o'_i} + \mu) - \mathbf{w}||_2. \tag{14}$$

The gradient of this error with respect to $\mathbf{o'_i}$ is thus

$$E^{'}(\mathbf{o'_i})d\mathbf{o'_i} = 2\mathbf{W}^T V^T[V(\mathbf{W}\mathbf{o'_i} + \mu) - \mathbf{w}]. \tag{15}$$

This error function is quadratic and hence convex. To find the projection that minimizes $E(\mathbf{o'_i})$ we set the gradient to 0 and solve the linear system for $\mathbf{o'}$.

$$\mathbf{A}\mathbf{o'} = \mathbf{b} \tag{16}$$

where

$$\mathbf{A} = \mathbf{W}^T\mathbf{V}^T\mathbf{V}\mathbf{W} \tag{17}$$

and

$$\mathbf{b} = \mathbf{W}^T\mathbf{V}^T(\mathbf{w} - \mathbf{V}\mu). \tag{18}$$

Once we have obtained our projection estimate $\mathbf{o'}$ we can reconstruct the object using equation 6. This reconstructed object, $\hat{\mathbf{o}}$, will be complete in that it minimizes the error between itself and the known portion $\mathbf{o_p}$ while predicting the unknown portions of $\mathbf{o_p}$.

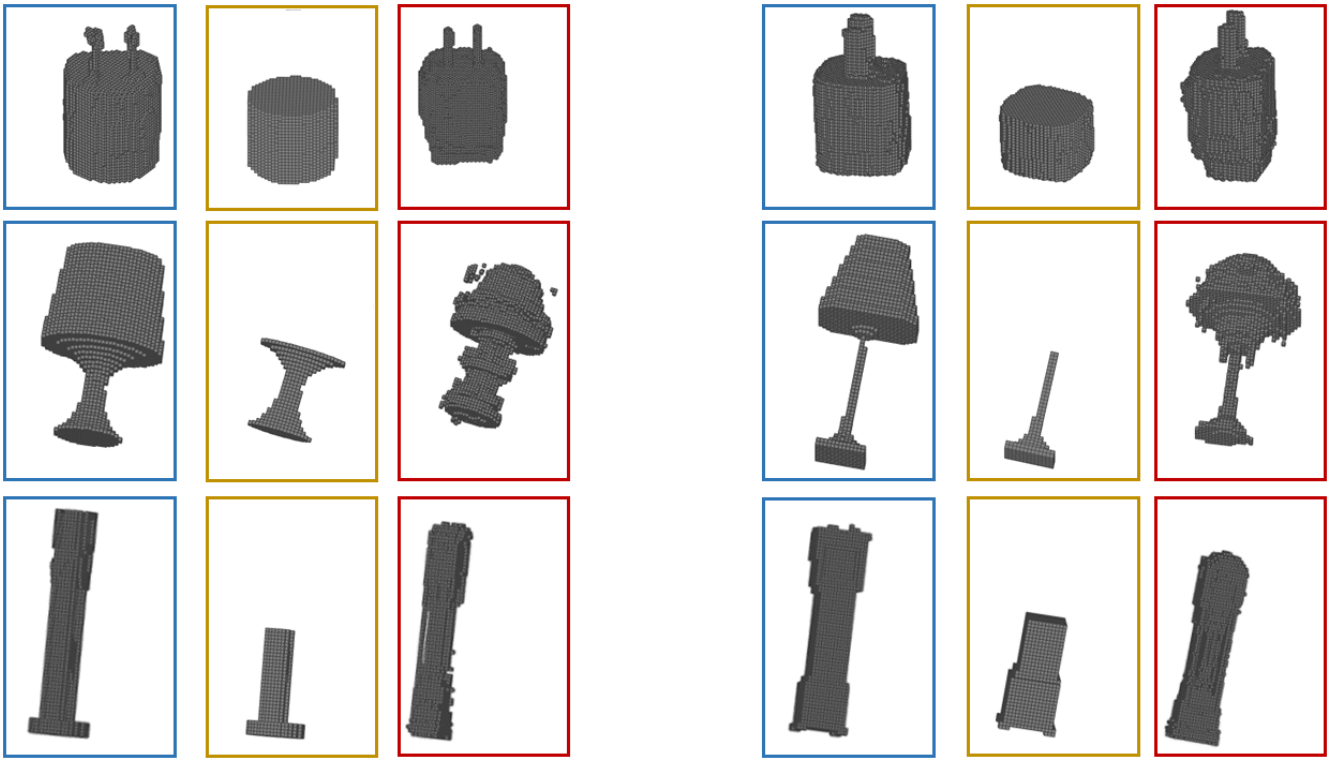

Fig. 2. Partial object completion. Original object is shown in blue boxes (left), the partial queries are shown in gold (middle), and the resulting estimated object is shown in red (right).

IV. Experimental Results

In order to characterize the behavior of our approach, we conducted experiments on three datasets. The first dataset is comprised of 20 wall USB charging plugs. These plugs were aligned to canonical orientation with the prongs directed upward and scanned on a MakerBot 3D scanner. The scanned mesh files were then voxelized into $254 \times 254 \times 254$ objects, forming the dataset. The other two datasets were both sourced from ShapeNet [7]. One consists of a random sampling of standing grandfather clocks while the other contains a random sampling of tabletop lamps. ShapeNet provides an appealing source of data because the objects it contains are oriented and scaled to canonical pose and size. Both the lamp and clock datasets were randomly split into two sub-groups for training and testing. The lamp training set consists of 40 lamps while the lamp test set consists of 10. The clock training set consists of 35 clocks and the clock test set consists of 8. Both lamps and clocks consist of size $273 \times 273 \times 273$ objects. Both the clock and lamp class manifolds were learned with 17 components while the plugs used 10.

A. Partial Object Completion

Each of our three datasets were used to complete partially specified objects. The plug dataset illustrates the case where the partial object is not novel; it was present in the group of objects used to find the object-class manifold. The clock and lamp sets were used to examine the novel case; the basis for each class was created from a separate training set while reconstruction of partially specified objects was performed on a separate pool of novel objects. A random subset of plugs, and each clock and lamp in the test set, were selected to be reconstructed from a partial fragment. Each of these objects was cut in half and the resulting incomplete object was used as input to our completion algorithm. Figure 2 illustrates some of these results. The right portion of the figure illustrates relatively successful reconstructions. While the results usually are not perfect and may contain significant noise, the overall shape profile remains intact. The left portion of Figure 2 illustrates more challenging cases where the resulting reconstruction differs in more meaningful ways from the original. Generally, this results when the query object has novel portions not present in the training objects. We observed that the lamp dataset was by far the most challenging from a reconstruction standpoint as it exhibits significant intra-class variance. Such classes will require a larger number of training objects and basis components for high quality results than more homogeneous classes.

B. Pose Estimation

We next examined our approach's ability to determine the pose of objects. To reduce the dimensionality of the problem, we assume that objects are centered in a reference frame and are upright. To fully specify an object's pose under such circumstances, we require an estimate of that object's rotation about its vertical axis. We tested the results of our pose estimator on all three datasets. Each object's pose was

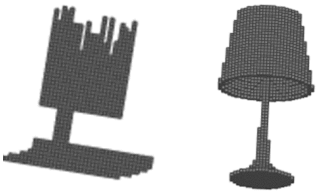

Fig. 3. Left: The lamp with successful pose estimation. Right: A lamp with incorrectly estimated pose.

Test Set	Plug Accuracy	Clock Accuracy	Lamp Accuracy
Complete Objects	1.0	1.00	0.1
Partial Objects (Top Half)	0.95	0.95	0.6
Partial Objects (Bottom Half)	0.95	0.88	0.4

TABLE I
POSE ESTIMATION RESULTS

Test Set	Clock Precision	Clock Recall	Lamp Precision	Lamp recall	Combined Accuracy
Complete Objects	0.73	1.00	1.00	0.70	**0.83**
Partial Objects (Top Half)	1.00	1.00	1.00	1.00	**1.00**
Partial Objects (Bottom Half)	N/A	0.00	0.56	1.00	**0.56**

TABLE II
CLASSIFICATION RESULTS ON THE CLOCK AND LAMP TEST SET

estimated by rotating it, in intervals of 36 degrees, about its vertical axis. We evaluate each candidate rotation as described in section III-B and select the highest scoring pose as our estimate. The results of this experiment are summarized in Table I. While we achieved perfect pose estimation accuracy on the plug and clock datasets, we observed poor results on the lamps. Because most lamps exhibit a high degree of rotation symmetry about their vertical axis, this set is intrinsically very challenging. Figure 3 illustrates the single success and an example failure case on the lamp dataset; the failed lamp is almost completely symmetrical with only tiny texture variations to distinguish its orientation while the successfully localized lamp has much more distinct variation about its vertical axis. Interestingly, both top-only and bottom-only lamp experiments yielded significantly better results than the complete object experiment. When analyzing this more closely, we discovered that the top-only and bottom-only experiments experienced success on different lamps. This is likely the result of conflicting cues between the top and bottom portions of lamps; for some lamps the base provided misleading results while on others, the top was misleading.

C. Classification

Using the test set of 10 lamps and 8 clocks, we performed a two class classification experiment. We ran three types of experiments. The first experiment used the full test object as input to our classifier while the second and third experiments applied the approach outlined in section III-D to partial objects. In experiment two, we used only the top half of each object (simulating an occluded top) while in experiment three we used only the bottom half. While our classifier achieved good performance on full objects, it remarkably achieved perfect performance when given only the top half of objects and poor performance when given only the bottom half. The most reasonable explanation is that the tops of lamps look quite different from the tops of clocks while their bases can look similar. As long as the non-occluded portion of the query object is somewhat distinct, our approach is able to achieve good performance, even when much of the rest of the object is missing. It is also important to note that the classifier used in all three experiments was the same; we did not retrain it based on partially occluded objects for experiments two and three. Although the classifier was only trained on on full objects, it was able to generalize to distance field differences obtained from partial objects.

V. CONCLUSION

Despite the ubiquity of object-centric tasks in modern robotic applications, modeling, storing, reasoning about, and extrapolating from previously encountered objects to novel partially observed objects is still an open problem. We propose the use of Variational Bayesian Principal Component Analysis as the basis for a multi-class object representation. Novel objects can be localized, classified, and completed by projecting them onto class basis and then projecting back into object space.

A key advantage of our approach to object representation is its ability to perform partial object completion. Because objects in real environments are rarely observable in their entirety from a single vantage point, the ability to produce even a rough estimate of the hidden regions of a novel object can be extremely useful. Furthermore, being able to classify partial objects dramatically improves the efficiency of object-search tasks by not requiring the agent to examine all candidate objects from multiple viewpoints. Experimentally, we illustrated our method on several datasets consisting of scanned objects. We were able to successfully estimate the rotational pose of novel objects, reconstruct partially unobserved objects, and categorize novel objects by class. We also showed that not only could partial objects can be completed, but that these completions are of suitable quality for use in classification.

References

[1] A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *The Journal of Machine Learning Research*, 11:1957–2000, 2010.

[2] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 998–1005, 2010.

[3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[4] C. M. Bishop. Variational principal components. In *International Conference on Artificial Neural Networks*, pages 509–514, 1999.

[5] C. M. Bishop. Bayesian PCA. In *Advances in Neural Information Processing Systems*, pages 382–388, 1999.

[6] C. R. Maurer, Jr., R. Qi, and V. Raghavan. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:265–270, 2003.

[7] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[8] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert. Parts-based 3D object classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 82–89, 2004.

[9] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[10] D.G. Lowe. Local feature view clustering for 3D object recognition. In *IEEE International Conference on Computer Vision*, volume 1, pages 682–688, 2001.

[11] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.

[12] L. Nan, Ke Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)*, 31, 2012.

[13] Y. Li, A. Dai, L. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3D reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446, 2015.

[14] M. Brown and D.G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *International Conference on 3-D Digital Imaging and Modeling*, pages 56–63, 2005.

[15] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61:611–622, 1999.

[16] M. Pontil and A. Verri. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[17] S. Marini, S. Biasotti, and B. Falcidieno. Partial matching by structural descriptors. In *Content-Based Retrieval, number 06171 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl*, 2006.

[18] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[19] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3D object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33:33 – 80, 1986.

[20] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2155–2162, 2010.

[21] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, and A.J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[23] V. Nair and G. E. Hinton. 3D object recognition with deep belief nets. In *Advances in Neural Information Processing Systems*, pages 1339–1347, 2009.

[24] Y/ Kim, N. J. Mitra, D. M. Yan, and L. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics*, 31:138:1–138:11, 2012.

[25] Y. Kim, N. J. Mitra, Q. Huang, and L. Guibas. Guided real-time scanning of indoor objects. In *Computer Graphics Forum*, volume 32, pages 177–186, 2013.