Bayesian Eigenobjects: A Unified Framework for 3D Robot Perception

Benjamin Burchfiel† and George Konidaris⋆
†Duke University, Durham NC
⋆Brown University, Providence RI
bcburch@cs.duke.edu, gdk@cs.brown.edu

***Abstract*—We introduce Bayesian Eigenobjects (BEOs), a novel object representation that is the first technique able to perform joint classification, pose estimation, and 3D geometric completion on previously unencountered and partially observed query objects. BEOs employ Variational Bayesian Principal Component Analysis (VBPCA) directly on 3D object representations to create generative and compact probabilistic models for classes of 3D objects. Using only depth information, we significantly outperform the current state-of-the-art method for joint classification and 3D completion in both accuracy and query time. Additionally, we show that BEOs are well suited for the extremely challenging task of joint classification, completion, and pose estimation on a large dataset of household objects.**

I. Introduction

Robot-object interaction requires several key perceptual building blocks including object pose estimation, object classification, and partial-object completion. These tasks form the perceptual foundation for many higher level operations including object manipulation and world-state estimation.

In real-world settings, robots will inevitably be required to interact with previously unseen objects. While databases with tens of thousands of object models exist, the world contains orders of magnitude more variation; it is impractical to assume that a robot operating in an unstructured environment has a model of every object it may encounter. Nevertheless, the common practice in robotics is still to build a library of 3D object models and match them to encountered objects in the world, often using the iterative closest point method (ICP) [21], to obtain a pose estimate or class. While this can be successful in highly controlled scenarios where an exact model of the object exists in the library, it will fail in less controlled environments containing a wide variety of objects. Consider a robot designed to clear dishes off of a table. The variation in the size and shape of bowls, platters, and plates that the robot could encounter is huge. While such a task might be feasible for a specific set of dishes, creating a general purpose table clearer is currently beyond our abilities; new approaches are required to allow for generalization across highly variable objects.

This work focuses on the joint classification, pose estimation, and geometric completion of previously unencountered objects and introduces a novel approach, Bayesian Eigenobjects (BEOs), that is well suited to these tasks. BEOs use Variational Bayesian Principal Component Analysis (VBPCA) as the basis for a multi-class object representation. By learning a compact basis for each class, we are able to represent objects using their projection coefficients. With this representation, novel objects can be localized, classified, and completed by projecting them onto class bases and then projecting back into object-space. Our method scales gracefully with the number of objects in each class, requiring constant per-class computation for projection and reconstruction even as the number of previously encountered objects increases.

A key feature of our single, unified, object representation is the ability to perform partial-object completion in 3D. Because objects in real environments are rarely completely visible 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. We applied our method to a dataset of common 3D objects to evaluate its ability to perform pose estimation, classification, and 3D completion. We were able to successfully estimate the rotational pose of novel objects, reconstruct partially unobserved objects, and categorize the class of novel objects. Our classification results on 3D objects significantly outperforms the state-of-the-art competing approach to joint classification and geometric completion in apples-to-apples comparisons, is significantly faster, and can also jointly estimate pose.

II. Background

A. 3D Object Representation and Classification

While considerable work has been performed on the recognition of known 3D objects [33, 23], less progress has been made on representing classes of 3D objects in a general way. One approach [37, 36, 16] is to construct a large database of complete and high quality object scans. When a novel object is encountered, a query is performed to find the most similar known object. 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. These approaches suffer however, if an exact match is not found. While some approaches still attempt to find a nearest match in such a case [16, 3], 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, and will yield poor results if coverage of the space is insufficient. Database methods also have scaling issues; the size of the database increases with the number of training examples.

Parts-based approaches learn a dictionary of parts and represent objects as a combination of dictionary elements [8, 18]. Parts-based approaches have the advantage of 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. However, 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 not necessarily an issue for recognition, this is a limitation for other tasks including object interaction. Additionally, object correspondence makes parts-based approaches impractical for representing diverse classes of objects.

The current state-of-the-art in the area consists of deep convolutional neural network (CNN) methods. These can be broken down into two categories, voxel-based representations [22, 27, 35] and multi-view representations [29, 4], with some methods attempting to combine the two [12]. Voxel representations represent objects explicitly in 3D, as a collection of discretized volumetric pixels (voxels), while multi-view approaches represent 3D objects as a collection of 2D projections. Despite these significantly different approaches, recent classification performance has been quite competitive between the two approaches [22].

B. Pose Estimation

Existing pose estimation methods employ a variety of techniques including image contours and shape templates [19] as well as spherical surface features (spin images) [14]. ICP [25] is commonly used for pose estimation when a 3D model of the object is known. Because ICP is vulnerable to local minima, it sometimes produces dramatically incorrect results. More recent approaches have used deep neural networks (DNNs) for for joint detection and pose estimation, such as Tulsiani and Malik [31], and have also shown promising results, as have some probabilistic localization methods [13]. Nevertheless, all of these methods have focused on known objects, not the more challenging task of estimating the canonical pose of a novel query object.

Methods that do aim to predict category-pose of novel objects are rare. One approach, Elhoseiny et al. [11], employs a discriminative multi-view CNN able to jointly perform pose estimation (into one of sixteen bins) and classification.

C. 3D Object Completion

While object completion for small holes in surface geometry has been fairly successful [2], inferring shape in large missing portions of a novel object remains an active area of research. One approach is to build a database of single viewpoint depth images corresponding to known 3D models and, given a query, extract information from the closest match to reconstruct the novel object [24]. The current state-of-the-art in this area trains CNNs to predict surface normals [32] or full 3D structure [35] from 2.5D images. A drawback of these techniques is poor scaling with respect to object resolution; current methods only scale to objects on the order of size 30^3. Other approaches exist which aim to provide 3D completion and classification [35], [34], although only Wu et al. [35] uses a single representation for both tasks and is able to do partial-object classification. Amodal perception approaches have also been proposed [28]. While these methods do not provide detailed 3D completions, they do estimate a coarse 3D bounding box from a 2.5D image. The current state-of-the-art representation for combined 3D completion and classification is 3DShapeNets [35], which learns a deep belief network representation of 3D objects enabling joint classification and 3D completion of partially observed query objects. While parts-based models struggle with modeling large variations across multiple diverse classes and symmetry-based approaches struggle, unsurprisingly, on asymmetrical objects, 3DShapeNets generativly models shape distributions with little assumption about the underlying geometric properties of the data.

D. 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) [17] which models each datapoint, $\mathbf{x_i}$, as

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

where $\mathbf{X}$ is a matrix containing all datapoints such that column i of $\mathbf{X}$ is $\mathbf{x_i}$, $\mathbf{W}$ is a basis matrix, $\mathbf{c_i}$ is the projection of $\mathbf{x_i}$ onto that matrix, μ is the mean of all datapoints, and $\varepsilon_\mathbf{i}$ is zero mean Gaussian noise associated with datapoint i. PPCA also makes the explicit assumption that each projected datapoint, $\mathbf{c_i}$, is generated from a zero mean Gaussian distribution. The model parameters for PPCA may be efficiently estimated using EM, which alternates between updating the estimate of each datapoint's coefficient, $\mathbf{c_i}$, and updating $\mathbf{W}$, μ, and ε. This probabilistic approach to PCA is well suited to density estimation and data compression. Bayesian PCA (BPCA) [6] further extends this model by introducing (Gaussian) priors (parametrized by $\mathscr{H}$) over the elements of μ and W. This allows BPCA to model the entire posterior probability of model parameters:

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

Note that column i of $\mathbf{C}$ is simply $\mathbf{c_i}$. Unfortunately, there is no analytic form for this probability so the straightforward application of EM is problematic. VBPCA overcomes this by requiring the posterior to have a factorized form. As a result, each factor can be iteratively updated separately during optimization, with the others held constant. VBPCA approximates the posterior probability as:

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

where $q(\mathbf{W}, \mu, \mathbf{C})$ is a factored posterior approximation:

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

This approximation allows us to use EM to perform VPBCA in the same way it is employed for PPCA. For a more detailed explanation, please refer to Bishop [5].

VBPCA can be conceptualized as a probabilistically 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. $n \ll d$ [5], 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 enable inference about novel partial-object queries.

A. Class Models: Eigenobject Construction via VBPCA

BEOs are learned from a library of known objects of several classes, with each object consisting of a complete 3D scan. Each object is converted into a 3D voxel-based representation with a canonical orientation. Class models are learned by vectorizing objects in each class and extracting a low-dimensional class subspace, $\mathbf{W_s}$, using VBPCA.

Note that s denotes a single class and S is the set of all classes. After training, a novel object, $\mathbf{o}$, can be projected onto $\mathbf{W_s}$ via

$$\mathbf{o_s}' = \mathbf{W_s}^T(\mathbf{o} - \mu_s) \tag{5}$$

where μ_s is the mean of all training objects in class s. Conversely, any point in the space of $\mathbf{W_s}$ can be converted back to a voxel object via

$$\hat{\mathbf{o}}_\mathbf{s} = \mathbf{W_s}\mathbf{o_s}' + \mu_s. \tag{6}$$

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

We need not store or query an entire object database; instead, we need only store $\mathbf{W_s}$ and μ_s for each class of objects ($\forall s \in S$). We can also represent any object in a given class using a single coefficient vector of dimension k_p. In practice, the number of basis for each class is far less than the dimensionality of each datapoint ($k_s \ll d$), providing a compact representation.

B. Object Classification

An essential part of many robot tasks is novel-object classification. Let the learned models for multiple classes be denoted $\theta_1, \theta_2, ..., \theta_m$, where m is the number of classes and $\theta_s = \{\mathbf{W}_s, \mu_s\}$, and let the novel query object be denoted $\mathbf{o_q}$. We wish to estimate the class label, $\hat{l}_q$, of $\mathbf{o_q}$ by selecting $\hat{l}_q$ from set S.

Our classification method leverages the trained low-dimensional space to learn a compact density model for each class. While such a density model is infeasible in 3D space as even size 30^3 objects contain tens of thousands of variables, it is possible in this much smaller projected space where each class is modeled as an anisotropic Gaussian.

From the learned subspaces for each class, we construct a single larger subspace upon which objects across all classes lie:

$$\mathbf{W} = [\mathbf{W_1}, ..., \mathbf{W_m}, \mu_1, ..., \mu_m].$$

Note that $\mathbf{W}$ is a $d \times 2m$ matrix with rows corresponding to dimensions in voxel space and columns corresponding to basis vectors. $\mathbf{W}$ may contain dependent rows which will serve as additional variables to estimate without increasing the expressiveness of our model. To prevent this, we find a matrix, $\mathbf{W}'$, with orthonormal columns that span the same space as $\mathbf{W}$. This can be straightforwardly accomplished by letting $\mathbf{W}' = \mathbf{U}'$ where $\mathbf{USV^T} = \mathbf{W}$ is the singular value decomposition of $\mathbf{W}$ and $\mathbf{U}'$ is formed by retaining only the first $rank(\mathbf{W})$ rows of $\mathbf{U}$.

After learning the shared subspace, density models for each class can be found by estimating the mean and covariance of m multivariate Gaussian distributions, each of dimensionality $rank(\mathbf{W})$. Estimation of the mean is straightforward:

$$\bar{\mathbf{o}}'_\mathbf{s} = \sum_{\mathbf{o}'_\mathbf{s} \in \mathbf{O}'_\mathbf{s}} \frac{\mathbf{o}'_\mathbf{s}}{n_s}, \tag{7}$$

where n_s is the number of training objects in class s and $\mathbf{O}'_\mathbf{s}$ is the $rank(\mathbf{W})$ by n_s matrix of projected training objects in class s.

Unfortunately, estimating the population covariance is more difficult. The simplest approach uses the sample covariance matrix:

$$\Sigma_s = \frac{1}{n_s - 1}\bar{\mathbf{O}}'_\mathbf{s}\bar{\mathbf{O}}'^T_\mathbf{s},$$

where $\bar{\mathbf{O}}'_\mathbf{s}$ is created by subtracting $\bar{\mathbf{o}}'_\mathbf{s}$ from each column in $\mathbf{O}'_\mathbf{s}$. Unfortunately, this method works poorly in practice. Although $\mathbf{W}'$ is far smaller than full 3D object-space, it is still fairly large, easily several hundred dimensions. As a result, accurately estimating the covariance for each class with only several hundred datapoints points per class is problematic. We utilize a combination of two forms of covariance shrinkage to regularize the estimated covariance matrix. Starting with Σ_s, we first shrink it by adjusting its eigenvalues towards their means, finding Σ'_s. Next, we further regularize Σ'_s by regressing it towards Σ'_{s-2}, the so-called "two parameter covariance matrix". Diagonal elements of Σ'_{s-2} are all equal to the mean variance in Σ'_s while non-diagonal elements of Σ'_{s-2} are all equal to the mean off-diagonal covariance in Σ'_s. The final estimate for the covariance of a given class is thus

$$\Sigma''_s = \lambda\Sigma'_s + (1 - \lambda)\Sigma'_{s-2}. \tag{8}$$

The precise calculation of the optimal shrinkage amount, λ, and the eigenvalue shrinkage amounts used in Σ' are outside the scope of this paper, but require minimizing mean squared error (MMSE). Ledoit and Wolf [15], Daniels and Kass [9], and Schäfer and Strimmer [26] offer more detail.

Once probability density functions for each of the classes have been learned, classification can be performed in a maximum a posteriori fashion. Given query object $\mathbf{o_q}$, its projection,

$\mathbf{o'_q}$, onto $\mathbf{W}'$, and $P(s)$, the prior probability of class s, we can find the most probable class label, $\hat{l}_q$,

$$\hat{l}_q = \underset{s \in S}{\arg\max} \;\; \frac{P(s)D(\mathbf{o'_q}|s)}{\sum_{s_j \in C} D(\mathbf{o'_q}|s_j)}, \tag{9}$$

where $D(\mathbf{o'_q}|c_i)$ denotes the density of the learned PDF, found using equations 7 and 8, for class s at location $\mathbf{o'_q}$,

$$D(\mathbf{o'_q}|s) = \frac{1}{\sqrt{(2\pi)^a |\Sigma_s|}} exp\left(-\frac{1}{2}(\mathbf{o'_q} - \bar{\mathbf{o'_s}})^T {\Sigma_s}^{-1}(\mathbf{o'_q} - \bar{\mathbf{o'_s}})\right), \tag{10}$$

where a denotes $rank(\mathbf{W}')$ for readability.

Unlike methods that operate directly on objects, there is no need to tune 3D features for individual classes. Furthermore, our approach can accommodate significantly higher resolution objects then competing DNN methods; BEOs can handle objects of size 250^3, which contain over fifteen million voxels, while constructing a DNN to to process such high-dimensional input is infeasible.

C. Pose Estimation

Like classification, pose estimation is necessary for many robotic manipulation and planning tasks, especially object manipulation. While it is relatively straightforward to acquire a rough estimate of an object's position given an object detection and segmentation, determining orientation is more difficult. Here, we do not assume we have a model of the query object, making our pose estimation task far more difficult; we cannot simply match the query object to its exact model. We must instead determine its pose relative to a canonical class baseline. We accomplish this by employing a try-verify approach, also known as pose estimation by search [20]. These approaches use a generative model of the object and sample from various configurations to find one that is maximally likely or minimizes scene error.

Let $\mathbf{R}(\mathbf{o_q}) = \{\mathbf{o_q^1}, \mathbf{o_q^2}, ...\mathbf{o_q^p}\}$ be query object $\mathbf{o_q}$ in p orientations. To estimate the true orientation of $\mathbf{o_q}$, $\hat{r}_q$, one possible solution is

$$\hat{r}_q = \underset{\mathbf{o_q^r} \in \mathbf{R}(\mathbf{o_q})}{\arg\min} \;\; ||\mathbf{o_q} - \mathbf{o_q^r}||_2. \tag{11}$$

This can result in ambiguity however, particularly with only partially specified query objects. This estimator ignores learned class properties; a cabinet, for instance, might project well onto the space of toilets because toilets have a large rectangular back but such a toilet would be highly unusual. Using only equation 11 does nothing to address this, motivating an alternate approach, conceptually based on the density estimator, equation 10, used in classification:

$$\hat{r}_q = \underset{\mathbf{o_q^r} \in \mathbf{R}(\mathbf{o_q})}{\arg\max} \;\; \frac{P(r)D(\mathbf{o_q'^r}|s)}{\sum_{r_j \in R} P(r_j)D(\mathbf{o_q'^{r_j}}|s)}. \tag{12}$$

Here, $\mathbf{R}$ is the set of rotations/poses being searched over and $P(r)$ is the prior probability of pose r.[1]

Equation 12 selects the MAP pose estimate; an advantage of this estimator is its sensitivity to the misalignment of important geometry. Consider the example of a bathtub: while mostly symmetrical around their z-axis, bathtubs have a spout at one end which provides key information for pose estimation. An approach based on equation 11 weights error equally in all places while the density method can respect that some regions contain more information than others.

If a high resolution orientation is required, there may be a large number of candidate poses. Fortunately, each candidate rotation can be calculated independently and thus the process is straightforwardly parallel; it is possible to distribute the workload to multiple processors or accelerate it via GPU. Our experiments investigate both full 3DOF pose estimation as well as 1DOF rotation about the z axis.

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 laboratory environments, it is unreasonable to expect a robot will always 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 a query object may be. BEOs provide this ability because they offer a generative representation of objects in a class. Each learned basis provides an object-class manifold. If we can find the point on this manifold that best corresponds to a given partial object, projecting back into voxel object-space yields a prediction of the unobserved portions of the object.

Similar to Li et al. [16], we assume that the partial query objects consist of filled, empty, and unknown pixels. Let d' denote the number of known (filled and empty) elements of $\mathbf{o_q}$. It is useful to define a d' by d binary selection matrix V such that $\mathbf{V_q o_q} = \mathbf{w_q}$ where $\mathbf{w_q}$ is a length $d' < d$ vector consisting of only the known elements of $\mathbf{o_q}$. $\mathbf{V_q}$ can be created by constructing a size d identity matrix and then removing all rows corresponding to unknown elements of $\mathbf{o_q}$ (e.g. if the *ith* element of $\mathbf{o_q}$ is unknown then the *ith* row of the identity is removed). Let $\mathbf{o'_q}$ denote the smallest error projection of $\mathbf{o_q}$ onto basis $\mathbf{W}'$. The error induced by an arbitrary projection $\mathbf{o_q'^i}$ with respect to $\mathbf{o_q}$ is

$$E(\mathbf{o_q'^i}) = ||V_q(\mathbf{W}'\mathbf{o_q'^i}) - \mathbf{w_q}||_2^2. \tag{13}$$

The gradient of this error with respect to $\mathbf{o_q'^i}$ is thus

$$E^{'}(\mathbf{o_q'^i})d\mathbf{o_q'^i} = 2\mathbf{W'}^T V_q^T [V_q(\mathbf{W}'\mathbf{o_q'^i}) - \mathbf{w_q}]. \tag{14}$$

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

$$\mathbf{A_q o'_q} = \mathbf{b_q} \tag{15}$$

[1] We employ uniform priors in our experiments.

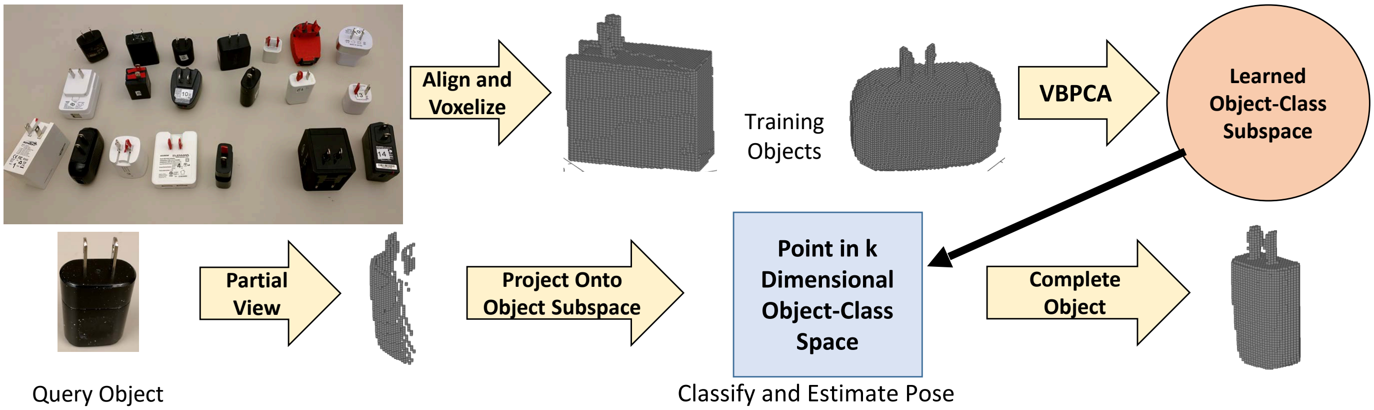
Align and Voxelize
Training Objects
VBPCA
Learned Object-Class Subspace
Partial View
Project Onto Object Subspace
Point in k Dimensional Object-Class Space
Complete Object
Query Object
Classify and Estimate Pose

Fig. 1: A high-level overview of the BEO process. Top: The training process. Bottom: An example query.

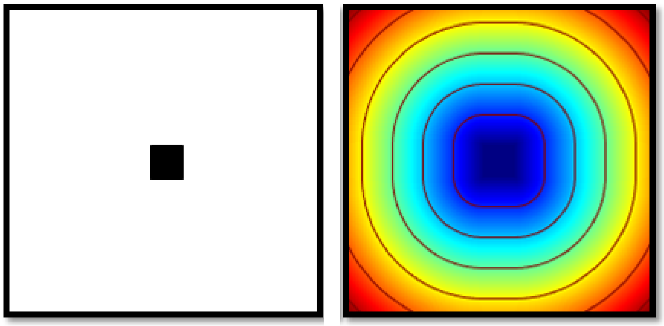

Fig. 2: Left: An example square 2D object Right: The resulting EDT (warmer colors are larger values)

where

$$\mathbf{A_q} = \mathbf{W'}^T \mathbf{V_q}^T \mathbf{V_q} \mathbf{W'} \tag{16}$$

and

$$\mathbf{b_q} = \mathbf{W'}^T \mathbf{V_q}^T \mathbf{w_q}. \tag{17}$$

When projecting for classification and pose estimation, we found it helpful in practice to employ extremely gentle lasso regularization [30] when solving for $\mathbf{o'_q}$,

$$\mathbf{o'_q} = \underset{\mathbf{o'^i_q}}{\arg\min} \; ||\mathbf{A_q} \mathbf{o'^i_q} - \mathbf{b_q}||_2 + \lambda ||\mathbf{o'^i_q}||_1 \ , \tag{18}$$

using $\lambda = 10^{-5}$.

Once we have obtained our projection estimate, $\mathbf{o'_q}$, we can complete the object using equation 6. Figure 1 illustrates this process. The completed object, $\mathbf{\hat{o}_q}$, minimizes the error between itself and the known portion $\mathbf{o_q}$ while predicting the unknown portions of $\mathbf{o_q}$. This process completes partially specified queries and performs joint classification and pose estimation with no additional modifications.

E. BEOs: Joint Pose, Class, and Geometry Estimation

BEOs can also be employed to perform all three operations (pose estimation, classification, and geometric completion) simultaneously. This full process consists of simultanious pose estimation and classification followed by completion and requires maximizing the joint probability over both pose and class. Because both classification and completion are quite sensitive to misalignment, we use a two step approach to pose estimation. In practice, equation 12 is unreliable when used to estimate global orientation for an object with unknown class. The search space is simply too large for good results. To address this, we employ a hierarchical coarse to fine approach. While we employ equation 12 during the finetuning phase, we employ a different method for the initial coarse estimate.

We define a coarse error based on the L_2 distance between an object and its back-projected version as,

$$e(\mathbf{o_q}, \mathbf{\hat{o}^r_q}) = 1 - \frac{||\mathbf{o_q} - \mathbf{\hat{o}^r_q}||_2}{|\mathbf{o_q}|}, \tag{19}$$

where a score of 1 denotes a perfect match and a score of 0 indicates that all voxels differ between the object and its back-projection. Intuitively, projecting an object onto its true class and proper orientation, and re-projecting it back into object-space, should result in a re-projection that closely matches the initial object. An initial estimator might simply find the class and orientation minimizing $e(\mathbf{o_q}, \mathbf{\hat{o}^r_q})$.

In practice, equation 19 works well for objects of the same class, but often fails when applied to objects of very different shape, making it unsuitable for comparing disparate objects across multiple classes.

We thus leverage a more nuanced representation of error. From $\mathbf{o_q}$ and $\mathbf{\hat{o}_q}$ we extract the Euclidean Distance Transform (EDT) [7] from each object, $\mathbf{D}$ and $\mathbf{\hat{D}}$ 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

Known Pose	Bathtub	Bed	Chair	Desk	Dresser	Monitor	Night Stand	Sofa	Table	Toilet	**Total**
BEO	48.0	**95.0**	**93.0**	**46.5**	64.0	**91.0**	55.8	**92.0**	**75.0**	80.0	**76.3**
Baseline	70.0	94.0	0.0	17.4	67.4	78.0	**75.6**	88.0	0.0	**82.0**	56.7
3DShapeNets [35]	**76.0**	77.0	38.0	22.1	**90.7**	74.0	38.4	57.0	1.0	79.0	54.4
Unknown Pose	Bathtub	Bed	Chair	Desk	Dresser	Monitor	Night Stand	Sofa	Table	Toilet	**Total**
BEO	4.0	64.0	83.0	16.3	51.2	86.0	36.0	49.0	76.0	46.0	54.5

TABLE I: Top: ModelNet10 classification accuracy (percent) with single-viewpoint queries comparing our results with 3DShapeNets and a baseline method. Bottom: Our method performing joint pose estimation, classification, and completion.

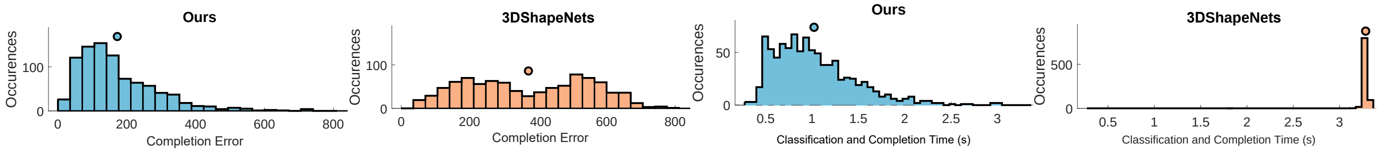

Fig. 3: Completion errors and query time for BEOs and 3DShapeNets. Mean error and time is indicated by circular dots.

create a more robust measure of distance:

$$e'(\mathbf{o_q}, \mathbf{\hat{o}_q^r}) = ||\mathbf{D} - \mathbf{\hat{D}_q^r}||_2. \tag{20}$$

Because it implicitly captures shape differences between two objects, this distance error provides a much more robust metric for comparison than equation 19. Refer to Figure 2 for an illustration of an example EDT.

During the first, coarse, step, we use EDT error from equation 20 to coarsely estimate the object's pose. Next, we finely discretize the rotation space in a region around the initial coarse estimate and rely on a modified version of equation 12,

$$\{\hat{r}_q, \mathbf{\hat{l}_q}\} = \underset{\mathbf{o_q^r} \in \mathbf{R}(\mathbf{o_q}),\ s \in S}{\arg\max} \frac{P(r) D(\mathbf{o_q'^r} | s) P(s)}{\sum_{r_j \in R} \sum_{s \in \mathbf{S}} P(r_j) D(\mathbf{o_q'^{r_j}} | s) P(s)}, \tag{21}$$

to obtain our final rotation estimate and classification label by marginalizing over both possible poses and possible classes. Note that $\mathbf{o_q'^r}$ denotes query object $\mathbf{o_q}$, in pose r, projected onto W'. Because equation 21 is unreliable when used to estimate global orientation but performs well in a region close to the correct pose, employing it only during the fine-tuning step proves effective.

IV. Experimental Results

We characterize the performance of our approach using the ModelNet10 dataset [35]. ModelNet10 consists of 4889 (3991 training and 908 test) aligned 3D objects, each of size 30^3, spread across 10 classes: {Bathtubs, Beds, Chairs, Desks, Dressers, Monitors, Night Stands, Sofas, Tables, Toilets}. Partial-views were obtained looking down along the z-axis of each object. We automatically selected a basis size capturing 60 percent of variance, between 30 and 70 components per class, and used zero-mean unit-variance Gaussian hyperparameters for regularization. We also illustrate some example high-resolution completions obtained from 20 USB charging plugs which were manually scanned in our lab using a MakerBot 3D scanner. These plugs were voxelized to size 254^3, forming an object-space with over 16.5 million dimensions, far larger than that accommodated by leading DNN methods such as 3DShapeNets [35].

We employed coordinate descent congealing [10] to roughly align the objects in each class, manually inspecting and refining the alignment as required. Some of our data was sourced from ModelNet [35], and arrived pre-aligned, while our manually scanned objects required alignment.

A. Classification and Object Completion

To provide a direct comparison with competing approaches, we assume the pose of each single-viewpoint query object is known and that the task consists of estimating the object's class label and full 3D geometry. We evaluate against the state-of-the-art existing method, 3DShapeNets [35], as well as a baseline which measures similarity to the mean training object in each class and selects the class with the most similar mean element. While other methods for performing single-view classification exist [4, 1, 12], they are incapable of providing 3D completion and so we focus on 3DShapeNets. Completion performance was measured by calculating the Euclidean error between the true geometry of the query object and the estimated completion. For each of the 908 test objects, test queries were created using a single top-down depth view. Table I summarizes the classification performance of each of these methods while Figure 3 contains their completion errors and query times. Note that the comparatively simple baseline method conflates several geometrically similar classes quite badly. Both 3DShapeNets and the baseline had significant trouble with tables. The top down view makes these challenging to distinguish from dressers and nightstands. Note that while ModelNet10 has been a fairly popular benchmark for full-object 3D classification, our experiments explore single-view classification performance, a significantly different task.

In both classification accuracy and reconstruction error,

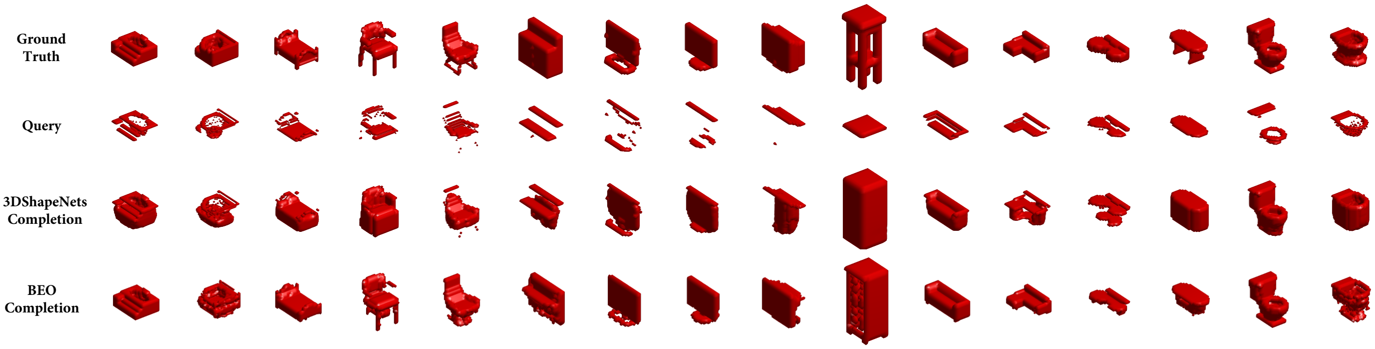

Fig. 4: A sampling of object completions. First row: The ground truth full object. Second row: query obtained from the novel object observed from a single view. Third row: Completions from 3DShapeNets. Fourth row: Completions using BEOs.

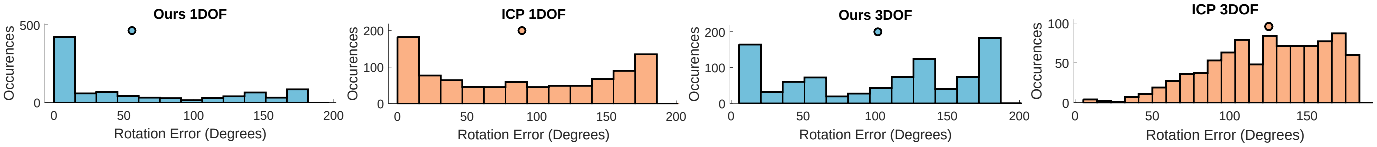

Fig. 5: Pose estimation error for BEOs and an ICP baseline. Mean error is indicated by circular dots.

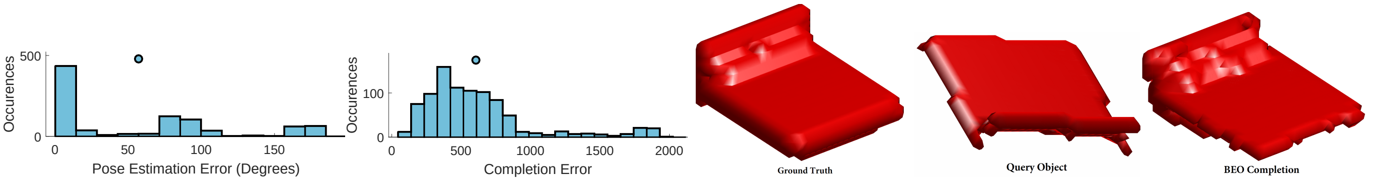

Fig. 6: Pose estimation and completion performance for BEO in the joint classification, pose estimation, and completion setting along with a successful query and completion.

BEO significantly outperforms 3DShapeNets, achieving nearly 20 percent greater classification accuracy. 3DShapeNets particularly struggled with tables, misclassifying them as night stands or dressers in nearly all instances due to their flat horizontal tops. While our BEO approach also exhibited this behavior to a far lesser degree, in many instances it was able to leverage the small differences in the size and aspect ratios of these objects to successfully classify them. Furthermore, although our query times exhibit some fluctuation because each query requires solving linear systems of differing size, our method is approximately three times faster than 3DShapeNets.

Our method differs significantly from 3DShapeNets for 3D completion. While 3DShapeNets classifies an object and then completes it, we perform object completion as part of our classification process. As a result, 3DShapeNets exhibits bimodal completion performance; when it misclassifies an object, its completion results degrade significantly. BEOs do not suffer from this drawback, often completing an unusual object (with respect to the training set) in a reasonable way, even while misclassifying it. Figure 4 illustrates some sample completions from BEOs and 3DShapeNets and is best viewed digitally with zoom.

B. Pose Estimation

To evaluate pose estimation performance, we performed experiments in both 1DOF with 1 degree of precision and 3*DOF* with 20 degrees of precision using the density estimator from equation 12. As 3DShapeNets cannot estimate pose, we compared against an ICP approach that warps the query object to the class-mean.

In 1DOF experiments, each query object was given a random orientation obtained by sampling uniformly at random from $[0, \pi)$. In 3DOF experiments, each query object's orientation was given by by sampling a quaternion uniformly at random from the surface of the 4D hypersphere. As above, queries consisted of a single viewpoint looking down along the z-axis. In both 1DOF and 3DOF, BEOs dramatically outperformed ICP.

C. Joint Pose, Class, and 3D Geometry Estimation

We next evaluated our entire procedure of joint pose estimation, classification, and 3D completion. Input query objects were obtained by randomly rotating objects about their z-axis and extracting a single top-down view. While BEO performance degraded somewhat in this more challenging instance, we achieve equal classification performance with 3DShapeNets without employing knowledge of the object's

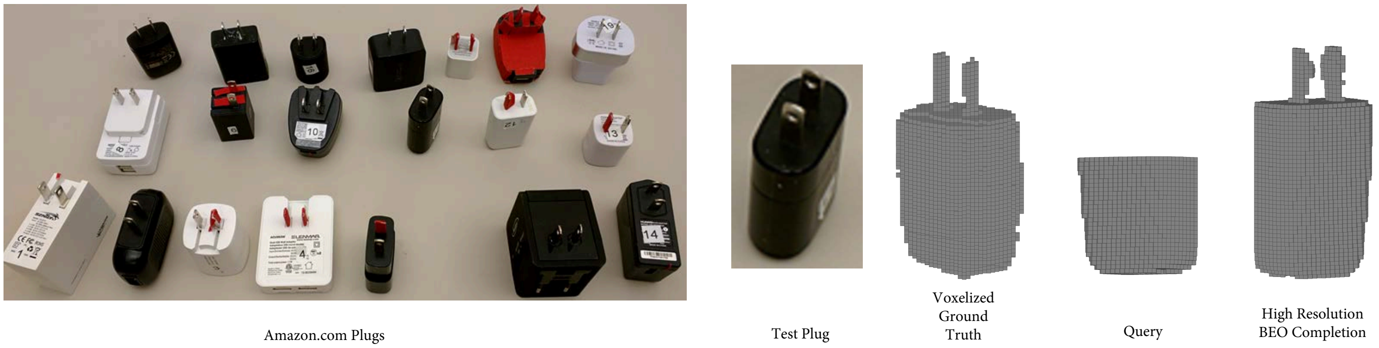

Fig. 7: An example completion using high-resolution manually scanned USB plugs.

pose. Table I contains our classification performance in this setting while Figure 6 shows our pose estimation and completion performance as well as an example query.

D. High Resolution Data

To demonstrate our approach applicability to high-resolution data, we obtained 20 USB wall plugs from Amazon.com and scanned them in our lab using a MakerBot 3D scanner. Each scan was then aligned and voxelized to size 254^3, a more than $60,000$ percent size increase over the ModelNet10 dataset. Due to the very limited number of objects, we trained on 19 of the plugs and evaluated completion performance on the remaining plug. Figure 7 illustrates an example completion with a voxelized visualization. At lower resolutions, recovering fine detail such as the shape of the prongs would be impossible.

V. Conclusion

A primary benefit of our 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 is mandatory. 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.

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 introduced Bayesian Eigenobjects, a method that uses Variational Bayesian Principal Component Analysis to construct a multi-class object representation. We showed that BEOs outperform the current state-of-the-art in joint classification and completion with queries of known pose, in both accuracy and classification performance, while also being significantly faster and scaling to higher resolution objects. Furthermore, BEOs are the first object representation that enables joint pose estimation, classification, and 3D completion of partially-observed novel objects with unknown orientations.

VI. Acknowledgments

The authors wish to thank the anonymous reviewers for their thorough feedback and helpful comments. This research was supported in part by DARPA under agreement number D15AP00104. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The content is solely the responsibility of the authors and does not necessarily represent the official views of DARPA.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[2] M. Attene. A lightweight approach to repairing digitized polygon meshes. *The Visual Computer*, 26(11):1393–1406, 2010.

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

[4] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki. Gift: A real-time and scalable 3D shape search engine. In *Computer Vision and Pattern Recognition*, June 2016.

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

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

[7] 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. *Pattern Analysis and Machine Intelligence*, 25:265–270, 2003.

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

[9] M. Daniels and R. Kass. Shrinkage estimators for covariance matrices. *Biometrics*, pages 1173–1184, 2001.

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

[11] M. Elhoseiny, T. El-Gaaly, A. Bakry, and A. Elgammal. Convolutional models for joint object categorization and pose estimation. *arXiv:1511.05175*, 2015.

[12] V. Hegde and R. Zadeh. Fusionnet: 3D object classification using multiple data representations. *rXiv:1607.05695*, 2016.

[13] J. Glover, R. Rusu, and G. Bradski. Monte Carlo pose estimation with quaternion kernels and the Bingham distribution. In *Robotics: Science and Systems*, 2011.

[14] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[15] O. Ledoit and M. Wolf. Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions. *Journal of Multivariate Analysis*, 139:360–384, 2015.

[16] 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.

[17] 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.

[18] S. Marini, S. Biasotti, and B. Falcidieno. Partial matching by structural descriptors. In *Content-Based Retrieval*, 2006.

[19] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *International Conference on Computer Vision*, pages 983–990, 2011.

[20] Venkatraman Narayanan and Maxim Likhachev. Perch: Perception via search for multi-object recognition and localization. In *International Conference on Robotics and Automation*, 2016.

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

[22] C. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *Computer Vision and Pattern Recognition*, 2016.

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

[24] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3D object shape from one depth image. In *Computer Vision and Pattern Recognition*, pages 2484–2493, 2015.

[25] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[26] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1):32, 2005.

[27] B. Shi, S. Bai, Z. Zhou, and X. Bai. Deeppano: Deep panoramic representation for 3-D shape recognition. *Signal Processing Letters*, 22(12):2339–2343, 2015.

[28] S. Song and J. Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *Computer Vision and Pattern Recognition*, 2016.

[29] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *International Conference on Computer Vision*, pages 945–953, 2015.

[30] R. Tibshirani. Regression shrinkage and selection via the lasso. *The Royal Statistical Society*, pages 267–288, 1996.

[31] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *2015 Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[32] S. Tulsiani, A. Kar, Q. Huang, J. Carreira, and J. Malik. Shape and symmetry induction for 3D objects. *CoRR*, abs/1511.07845, 2015.

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

[34] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[35] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[36] 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.

[37] 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.