Probablistic Object and Viewpoint Models for Active Object Recognition

Natasha Govender
MIAS (CSIR)
South Africa
ngovender@csir.co.za

Jonathan Warrell
MIAS (CSIR)
South Africa
jwarrell@csir.co.za

Philip Torr
University of Oxford
Brookes
United Kingdom
philiptorr@brookes.ac.uk

Fred Nicolls
University of Cape
Town
South Africa
fred.nicolls@uct.ac.za

Abstract— **For mobile robots to perform certain tasks in human environments, fast and accurate object verification and recognition is essential. Bayesian approaches to active object recognition have proved effective in a number of cases, allowing information across views to be integrated in a principled manner, and permitting a principled approach to data acquisition. Existing approaches however mostly rely on probabilistic models which make simplifying assumptions such as that features may be treated independently and that objects will appear without clutter at test time. We develop a number of probabilistic object and viewpoint models which are explicitly designed to cope with situations in which these assumptions fail, and show these to perform well in a Bayesian active recognition setting using test data in which objects appear in cluttered environments with significant occlusion.**

I. Introduction

3D Object recognition is an important task for mobile platforms to dynamically interact in human environments. This computer vision task also plays a fundamental role in the areas of automated surveillance, Simultaneous Localization and Mapping (SLAM) applications for robots and video retrieval. The recognition of objects that appear in cluttered environments with significant occlusions is a complicated and challenging problem. In this paper we present a feature-based Bayesian framework for integrating information in a principled manner across multiple views for active 3D object recognition. Experiments are conducted with different feature integration probability models on a challenging dataset.

In real-world situations multiple viewpoints are necessary for recognition [1] [2] as single viewpoints may be of poor quality and simply not contain sufficient information to reliably recognise or verify the object's identity unambiguously. This is especially true if they are occluded or appear in cluttered environments. Even with multiple views most systems have no tangible method of determining the accuracy of the recognition method. The system presented in this paper uses multiple images for 3D object recognition and provides a certainty/belief as to the current object's identity and pose.

Bayesian methods have proved effective in many active vision scenarios, and general frameworks for active sensing have been proposed [3], as well as specific models for scene exploration and tracking from surveillance videos [4] and object recognition from a mobile platform [5] [6]. In many of these cases however, attention is paid to the general problems of optimal methods for fusing data and planning sensing strategies while assuming a probabilistic model for the phenomenon of interest (object/environment) is given. By assuming simple probabilistic models and using highly controlled datasets, general methods for fusion and planning are easily demonstrated. However, it is unclear how well such models can cope in unconstrained settings.

In this paper we focus particularly on the question of how to represent an object probabilistically in order to perform effective active object recognition in the challenging scenario of highly cluttered test scenes. While adopting a standard Bayesian framework for data fusion, we explore two novel probabilistic models for representing object and viewpoint hypotheses in such circumstances and show these to perform well on a challenging dataset.

Our models build on a number of techniques that have proved effective for object recognition in both active and non-active settings. We rely firstly on interest points and local features extracted from training images. These have the advantage that their representation is more robust to occlusions, clutter and noise, and have successfully been used in 3D object recognition [7] [8] [9]. Our system uses the Scale Invariant Feature Transform (SIFT) [10] detector and descriptor to extract relevant object features. SIFT is robust to changes in illumination and affine transformation. Further, the features extracted from the training images are inputted directly into a vocabulary tree data structure [11] which facilitates quick matching and provides a method to discretize the feature space to reduce feature dimensionality when considered in the Bayesian framework. We note however that the probabilistic models we consider are not dependent on these particular low-level representation choices.

The structure of the paper is as follows. Section II discusses related work and Section III elaborates on the Bayesian active object recognition framework used. The probabilistic objects and viewpoint models are explained in Section IV with the experiments conducted described in Section V. Sections VI discusses the conclusions.

This work was funded by the Mobile Intelligent Autonomous Systems (MIAS) group in the Council for Scientific and Industrial Research (CSIR) South Africa.

II. Related Work

A wide range of general frameworks for active vision and active sensing have been explored, including information theoretic and Bayesian approaches [3] [6] [4] [5] [12], discriminative approaches [13] [14], and approaches based on other theoretical models such as possibilistic and Dempster-Shafner theory [15], [16]. We adopt a Bayesian framework due to its flexibility in incorporating diverse modeling choices in a principled manner. Further, in Borotschnig et. al. [16] a comparison was conducted between probabilistic (Bayesian), possibilitic and Dempster-Shafner theory approaches to data fusion. They concluded that the probablistic approach worked best for 3D active object recognition, although all these methods use test images with a single object in an uncluttered environment with no occlusions.

An early method to adopt a Bayesian approach [5] used an appearance based object representation namely a parametric eigenspace distribution, and updated the object and pose hypotheses using Bayes theorem. A limitation of this model is that, because a global eigenspace representation is used, the model copes poorly with recognizing highly occluded objects, and requires uncluttered test sequences.

A later method [13] uses SIFT to extract features from the training objects, which provides robustness to affine transformations and variable illumination. Using such features also avoids relying on global information for recognition. However, the method does not model the geometric structure of features, which is a disadvantage in confronting cluttered scenes (the test data are all uncluttered). Further, since the model is non-probabilistic, there is not a natural way to build additional assumptions into the framework.

Our framework is most directly related to that of [12]. This method is also based on SIFT, and, drawing on the techniques of [10] [7] for non-active recognition, incorporates geometric structure by filtering the features processed at a given view using the Hough transform to identify the most likely transformation from a training example. However, the method does not explicitly include the transformation as part of the probabilistic model, and further does not model the background or occlusion process. We deal with these issues by introducing a background distribution and latent occlusion and transformation variables, and incorporate a distribution over both object and pose, unlike [12].

Other related active recognition methods include Callari et. al. [6], who estimate Bayesian probabilities with neural nets and minimizes the expected ambiguity measured by Shannon entropy. They provide a measure for the object's idenitiy but not for pose. Also, the boosting and support vector machines strategy of [14] estimates both object and pose, but uses substantially uncluttered test data.

Following [12], we also make use of a vocabulary tree representation [11] for the low-level features in our models. This has shown to be effective in both 2D and 3D recognition tasks [11], and also in Simultaneous Localization and Mapping (SLAM) approaches for matching similar images and for loop closure [17].

III. Bayesian Active Object Recognition

Problem Statement: The *active object recognition* task can be defined as follows. At training time, for each object $o = 1...O$ we capture set of images, one at each of a series of P regularly spaced training views around the object indexed by their viewing angle, for example $\theta \in \{0^\circ, 20^\circ, 40^\circ, ...340^\circ\} = \Theta$, $P = |\Theta|$. For simplicity, we consider only varying the viewing angle around one axis (e.g. vertical), although minimal changes are necessary to incorporate viewpoints from across a viewing sphere. We thus have a training image $I^{\text{train}}_{o,\theta}$ for each object/view pair.

At test time, we are presented again with one of the training objects, and must identify a) the object present $o^\star$, and possibly b) the orientation of the object, which may be specified by the training viewpoint $\theta^\star$ corresponding to a reference test view. We are allowed to capture images of the test object at a sequence of test views, $\delta_1, \delta_2, ... \in \{0^\circ, 20^\circ, 40^\circ, ...340^\circ\}$, where the angles δ_t can be in any order. We label the image corresponding to the t'th test view $I^{\text{test}}_{\delta_t}$, and treat $\delta_1 = 0^\circ$ as a reference view (i.e. $I^{\text{train}}_{o^\star,\theta^\star}$ will denote the training view we believe corresponds to $I^{\text{test}}_{\delta_1}$). An active object recognition algorithm is allowed to choose both the sequence of test views $\delta_1, \delta_2, ...$, and when to stop capturing further viewpoints and generate an output.

A number of possible approaches can be taken to the active object recognition task. Bayesian probability provides a principled framework within which to build algorithms, and below we give the general outline of two possible methods. Each requires us to specify a probabilistic model for objects or object viewpoints, and we look at specific options for these in Sec. IV. Each also requires a *viewpoint selection* strategy, which we leave unspecified, since our primary focus will be on comparing different object/viewpoint models.

Bayesian algorithm (A) with Object Models: In this case, we require a probability model for our image feature representation of image I, $\mathbf{f}_I$ given object o: $P(\mathbf{f}_I|o)$. At a given time-step t during test time, we are interested in estimating $P_t(o) = P(o|\mathbf{f}^{\text{test}}_{\delta_1}, ...\mathbf{f}^{\text{test}}_{\delta_t})$, that is, the probability of each object given the images we have seen so far (writing $\mathbf{f}^{\text{test}}_{\delta}$ for $\mathbf{f}_{I^{\text{test}}_{\delta}}$). Assuming the images seen to be independent samples from the object's probability model, we can estimate $P_t(o)$ recursively using Bayes theorem:

$$P_t(o) = \frac{P(\mathbf{f}^{\text{test}}_{\delta_t}|o)P_{t-1}(o)}{\sum_o P(\mathbf{f}^{\text{test}}_{\delta_t}|o)P_{t-1}(o)} \tag{1}$$

If we have no information prior to testing, setting $P_0(o) = 1/O$ is an appropriate initial distribution. This update mechanism can be combined with a number of next viewpoint selection strategies, and we discuss our choice of the latter in the experimentation. A possible stopping criteria is to cease capturing further views when $\max(P_t(o)) > \tau$, where τ is a threshold parameter, and output $o^\star = \text{argmax}(P_t(o))$.

Bayesian algorithm (B) with Object/Viewpoint Models: Algorithm A above assumes that the images we view at test time are generated independently given the test object o. In general, this will not be the case, since we expect there to be high correlations between the images we see at particular viewpoints. We can build this information into our approach by using separate probability models for each object/viewpoint combination: $P(\mathbf{f}|o,\theta)$. Now we are interested in estimating at each time-step t a distribution $P_t(o,\theta) = P(o,\theta|\mathbf{f}^{\text{test}}_{\delta_1}, ...\mathbf{f}^{\text{test}}_{\delta_t})$, where we denote by $P_t(o,\theta)$ the probability at time-step t that the test object is o and the viewpoint at the reference test angle $\delta_1 = 0^\circ$ corresponds to training view θ. Again, we can estimate this distribution recursively:

$$P_t(o,\theta) = \frac{P(\mathbf{f}^{\text{test}}_{\delta_t}|o,\theta+\delta_t)P_{t-1}(o,\theta)}{\sum_o P(\mathbf{f}^{\text{test}}_{\delta_t}|o,\theta+\delta_t)P_{t-1}(o,\theta)} \qquad (2)$$

where we note that the offsets δ_t are required to select the correct likelihood models to combine at time-step t. As in method A, a uniform prior can be selected for $P_0(o,\theta)$, and we leave discussion of the next viewpoint selection strategy until the experimentation. If we are primarily interested in identifying the correct test object, we can further calculate:

$$P_t(o) = \sum_\theta P_t(o,\theta) \qquad (3)$$

at each time step, and again stop when $\max(P_t(o)) > \tau$, outputting $o^\star = \text{argmax}(P_t(o))$ and $\theta^\star = \text{argmax}_\theta(P_t(o^\star,\theta))$.

IV. Probabilistic Objects and Viewpoint Models

We outline below a number of possible models that can be used for the likelihoods in Eqs. 1 and 2. We discuss three kinds of model, which incorporate increasing levels of structure. We are particularly interested in identifying objects which may be occluded at test time, as will be explored in the experimentation, and the final two options below explicitly build this into the generative model.

A. Independent Features

For our first likelihood model, we assume that we have access to a preprocessing method to extract a sparse set of visual words from each training/test image (as noted in the experimentation, we will use a vocabulary tree for this purpose). Letting $\mathcal{N} = \{1...N\}$ be the set of all visual words (the dictionary), and assuming initially for convenience all images contain the same number of words, M, we can represent training image $I^{\text{train}}_{o,\theta}$ by the vector $\mathbf{f}^{\text{ind}}_{o,\theta} \in \mathcal{N}^M$, where the ordering of entries in $\mathbf{f}^{\text{ind}}_{o,\theta}$ is generated by assuming a fixed strategy, such as top-left to bottom-right. For algorithm A above, we can estimate the per-object distribution for individual features based on whether we observe an individual feature associated with an object during training:

$$\begin{aligned} P(n|o) \quad \propto \quad & p_a[(\sum_{\theta,m} f^{\text{ind}}_{o,\theta}(m) = n) = 0] + \\ & p_b[(\sum_{\theta,m} f^{\text{ind}}_{o,\theta}(m) = n) > 0] \end{aligned} \qquad (4)$$

where $n \in \mathcal{N}$ is a particular visual word, $[.] = 1$ for a true condition and 0 otherwise (the Iverson bracket), and p_a and p_b are parameters of the distribution controlling the probabilities when that node n is not seen and is seen respectively (which relate to the p_o and p_{no} parameters in [12]). The likelihood for a test image with features $\mathbf{f}^{\text{ind}}$ is then formed simply by treating all observed visual words as independent draws from Eq. 4:

$$P(\mathbf{f}^{\text{ind}}|o) = \prod_{m=1...M} P(f^{\text{ind}}(m)|o). \qquad (5)$$

An object viewpoint model for algorithm B can be formed similarly by storing $P(n|o,\theta)$ for each combination (removing the summations across θ in Eq. 4), and using these to form $P(\mathbf{f}^{\text{ind}}|o,\theta)$ similarly to Eq. 5. Finally, we note that, although we assume each image to contain M features, we can simply build a dependence on M into Eq. 5: $P(\mathbf{f}^{\text{ind}}|o) = P(M_{\mathbf{f}^{\text{ind}}}|o)\prod_{m=1...M_{\mathbf{f}^{\text{ind}}}} P(f^{\text{ind}}(m)|o)$, where $M_{\mathbf{f}^{\text{ind}}}$ is the length of $\mathbf{f}^{\text{ind}}$. If we assume $P(M_{\mathbf{f}^{\text{ind}}}|o)$ to be uniform within certain bounds (e.g. always between 10-1000 features) these factors will cancel in the Bayesian updates.

B. Binary Model

The independent features model above does not represent geometric structure in any way, and as such is susceptible to noise. This will especially be a problem in our experimental setup, in which we are interested in recognizing objects amongst clutter. We thus present here a simple likelihood model which embeds a notion of geometric structure. Again, we assume we have access to a preprocessed representation of each image containing as a sparse set of visual words in $\mathcal{N}$. We will also assume we have access to a geometric matching method which, for two images, generates a set of visual word correspondences for each of a set of geometric transformations, $t \in \mathcal{T} = \{1...T\}$ (which we assume to be discretized). In particular, we will use a Hough voting method similar to that described in [7]. As an illustration of how this works, consider the case that our transformations contain only x-y translations, t_x, and we denote the visual words from image 1 and 2 as $n^1_{1...M}$ and $n^2_{1...M}$ respectively, with (x,y) co-ordinates $x^1_{1...M}$ and $x^2_{1...M}$. For each pair of matching words across images such that $n^1_{m_1} = n^2_{m_2}$, a vote is added to the transformation $t_{x^2_{m_2}-x^1_{m_1}}$. The visual word correspondences for a given transformation are then all those pairs that voted for that transformation, and we can write $H_t(I_1,I_2)$ for the number of votes for transformation t between images I_1 and I_2. Although we considered only x-y translations above, the same scheme can incorporate scaling and rotation transformations if we store this information with the visual word representation (as for example when using quantized SIFT features).

With these preliminaries, for the case of a binary model with algorithm A, we set $\mathbf{f}^{\text{bin}}$ to be a binary indicator vector, $\mathbf{f}^{\text{bin}} \in \mathbb{B}^O$, where $\mathbb{B} = \{0,1\}$, and we have $|\mathbf{f}^{\text{bin}}| = 1$ (i.e. there is a single 1, and $O-1$ 0's). The position of the 1 in $\mathbf{f}^{\text{bin}}$ indicates the object model with the highest number of matching words after applying the chosen geometric matching procedure. Explicitly, denoting by $H_{o,\theta}(I)$ the maximum number of matches for a transformation between image I and training image $I^{\text{train}}_{o,\theta}$, which in the case of the Hough method above is $H_{o,\theta}(I) = \max_t H_t(I, I^{\text{train}}_{o,\theta})$, we can write:

$$\mathbf{f}^{\text{bin}}_I(o) = \begin{cases} 1 & \text{if } o = \text{argmax}_{o'}\, \text{argmax}_{\theta}\, H_{o',\theta}(I) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where ties are broken arbitrarily. Our likelihood model then assumes a simple form depending on a single parameter, p_c:

$$P(\mathbf{f}^{\text{bin}}|o) = p_c[\mathbf{f}(o) = 1] + ((1-p_c)/(O-1))[\mathbf{f}(o) = 0] \tag{7}$$

That is, we assume an object o generates a binary vector with $\mathbf{f}^{\text{bin}}(o) = 1$ with probability p_c, and a vector with a 1 positioned elsewhere with probability $1 - p_c$, this probability being divided evenly between the $O-1$ other cases.

A similar likelihood model can be formed for algorithm B. Here, let $\mathbf{f}^{\text{bin}} = (\mathbf{f}^{\text{bin,obj}}, \mathbf{f}^{\text{bin,pose}})$, $\mathbf{f}^{\text{bin,obj}} \in \mathbb{B}^O$, $\mathbf{f}^{\text{bin,pose}} \in \mathbb{B}^P$. Then, we set $\mathbf{f}^{\text{bin,obj}}(o)$ similarly to Eq. 6 and:

$$\mathbf{f}^{\text{bin,pose}}_I(\theta) = \begin{cases} 1 & \text{if } \theta = \text{argmax}_{\theta'}\, \text{argmax}_{o}\, H_{o,\theta'}(I) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The likelihood model is then:

$$\begin{aligned} P(\mathbf{f}^{\text{bin}}|o,\theta) \quad = \quad & p_c[\mathbf{f}^{\text{bin,obj}}(o) = 1 \wedge \mathbf{f}^{\text{bin,pose}}(\theta) = 1] + \\ & ((1-p_c)/(O \cdot P - 1))[\mathbf{f}^{\text{bin,obj}}(o) = 0 \vee \\ & \mathbf{f}^{\text{bin,pose}}(\theta) = 0] \end{aligned} \tag{9}$$

C. Occlusion Model

Although the model above incorporates geometric structure, it looses a large amount of information such as the number of matched and unmatched points as well as the extent of the background/clutter by projecting onto a binary feature vector. As a final model, we propose a likelihood function which more explicitly models the generative process of occluded test images. Here, we take $\mathbf{f}^{\text{occ}}$ to include the visual word indices as in the independent model, along with the discretized position and possibly scaling and rotation information of each word as required by the geometric matching process. For example, considering translation only, if we represent by $\mathcal{X}$ the set of all possible image positions, we let $\mathbf{f}^{\text{occ}} \in (\mathcal{N} \times \mathcal{X})^M$ (where the ordering of vector entries is arbitrary). We begin by outlining the object/viewpoint model for algorithm B, before mentioning how it is adapted for algorithm A.

Here, we will explicitly include transformation t as discussed above into the generative process. That is, for a given test image, t will be a latent variable. Further, we introduce a second set of latent variables $\alpha_{o,\theta} \in \mathbb{B}^{M_{o,\theta}}$, which represent occlusion maps for each of the training object/viewpoint images, where $M_{o,\theta}$ is the number of visual words in the training image for o and θ, and $\alpha_{o,\theta}(m) = 1$ implies that word m is visible in the test image, and 0 implies it is not. We then propose the likelihood model:

$$P(\mathbf{f}^{\text{occ}}|o,\theta) \quad = \quad \sum_{t,\alpha_{o,\theta}} P(t)P(\alpha_{o,\theta})P(\mathbf{f}^{\text{occ}}|o,\theta,t,\alpha_{o,\theta}) \tag{10}$$

Here, we may take a uniform distribution for $P(t)$, and simply characterize $P(\alpha_{o,\theta})$ as:

$$P(\alpha_{o,\theta}) = \prod_{m=1\ldots M_{o,\theta}} (\alpha_{o,\theta} p_d + (1-\alpha_{o,\theta})(1-p_d)) \tag{11}$$

where p_d is a general probability that a word is visible (which can be set from the rate of occlusion). Given test image representation $\mathbf{f}^{\text{occ}}$, object/viewpoint hypothesis o, θ and transformation t, we can construct a subset of matching visual words in the test image that are potential matches of training words, $\mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,t) \subset \{1...M\}$, which match in terms of visual word, and are transformed consistently according to t. For instance, for the Hough matching procedure described earlier, $\mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,t)$ contains all visual-word/location pairs in $\mathbf{f}^{\text{occ}}$ which voted for transformation t when matched with training image $I^{\text{train}}_{o,\theta}$. Only these visual words can be un-occluded. The remaining words must be generated by the background distribution, which we take to be uniform $p_e = 1/(N|\mathcal{X}|)$:

$$\begin{aligned} & P(\mathbf{f}^{\text{occ}}|o,\theta,t,\alpha_{o,\theta}) = \\ & \prod_{m=1\ldots M} ([\alpha_{o,\theta}(m) = 1][m \in \mathcal{M}(\mathbf{f}^{\text{occ}}|o,\theta,t)] + \\ & [\alpha_{o,\theta}(m) = 0]p_e) \end{aligned} \tag{12}$$

To avoid summing across all possible transformations in Eq. 10, we instead make the following approximation:

$$\begin{aligned} P(\mathbf{f}^{\text{occ}}|o,\theta) \quad & \approx \quad \tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta) \\ & = \quad \kappa_{o,\theta} \max_t \sum_{\alpha_{o,\theta}} P(\alpha_{o,\theta})P(\mathbf{f}^{\text{occ}}|o,\theta,t,\alpha_{o,\theta}) \end{aligned} \tag{13}$$

where $\kappa_{o,\theta}$ is a normalizing constant. This approximation implicitly assumes the likelihood is always highly peaked around the $t^\star$ achieving the maximum in Eq. 13.[1] If this is the case, $\kappa_{o,\theta} \approx 1$ for all (o,θ) and can be ignored. Also, assuming $p_d > 0.5$ and $p_d > p_e$, the maximization over t can be achieved by using the Hough transform [7]. Collecting terms, we can therefore further simplify the likelihood model to:

$$\begin{aligned} P(\mathbf{f}^{\text{occ}}|o,\theta) \quad \approx \quad & (p_d + (1-p_d)p_e)^{H_{o,\theta}(\mathbf{f}^{\text{occ}})} \cdot \\ & (1-p_d)^{M_{o,\theta} - H_{o,\theta}(\mathbf{f}^{\text{occ}})} p_e^{M - H_{o,\theta}(\mathbf{f}^{\text{occ}})} \end{aligned} \tag{14}$$

writing $H_{o,\theta}(\mathbf{f}^{\text{occ}})$ for $H_{o,\theta}(I)$ as introduced in Sec. IV-B, where $\mathbf{f}^{\text{occ}} = \mathbf{f}_I$. As in the independent features

[1] This assumption can be empirically tested. For our experimental set up, we tested it by plotting $P(\mathbf{f}^{\text{occ}}|o,\theta)$ for a large number of test images and (o,θ) combinations. The distributions were sharply unimodal in approximately 0.9 of the cases.

model, we can explicitly alter Eq. 14 to allow for a variable number of test features by letting $\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta) = P(M_{\mathbf{f}^{\text{occ}}}|o)\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta,M_{\mathbf{f}^{\text{occ}}})$, where $\tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta,M_{\mathbf{f}^{\text{occ}}})$ is as in Eq. 14, but with $M_{\mathbf{f}^{\text{occ}}}$ substituted for M. Again for a uniform $P(M_{\mathbf{f}^{\text{occ}}}|o)$ this does not affect the updates in Sec. III.

Finally, we can define an object likelihood for algorithm A in Sec. III by incorporating a further maximization across θ:

$$P(\mathbf{f}^{\text{occ}}|o) \propto \max_{\theta} \tilde{P}(\mathbf{f}^{\text{occ}}|o,\theta) \tag{15}$$

which can be evaluated as in Eq. 14 where θ is replaced by $\theta^{\star} = \text{argmax}_{\theta'} H_{o,\theta'}(\mathbf{f}^{\text{occ}})$.

V. Experiments

A. Dataset

For our experiments, we use the active recognition dataset introduced by [12]. The training data consists of everyday objects such as cereal boxes, ornaments, spice bottle etc. Images were captured every 20 degrees for each object against a plain background on a turntable using a Prosilica GE1900C camera. This means that there were 18 training images captured for each object in the database. For the test set, the same objects used in the training data were captured at every 20 degrees in a cluttered environment with significant occlusion by other objects in the dataset. For both training and test data, images are captured around the y-axis, which represents 1 degree-of-freedom (DoF). For our experiments, we choose a subset of 10 objects from the dataset for both testing and training, with minimal overlap between the learned and occluding objects. An example image from the test set is displayed in Figure 1.

B. Preprocessing using Vocabulary Tree

For our visual word representation, we use a vocabulary tree [11] learned in an unsupervised manner from SIFT features detected at interest points across all training images [10]. This is the same representation as used in [12].

The vocabulary tree is constructed using hierarchical k-means clustering where similar features are clustered together. k defines the number of children of each node of the tree. Initially, for the root of the tree, all the training data is grouped into k clusters. The training data is then used to construct k groups, where each group consists of SIFT descriptors closest to a particular cluster centre. This process is recursively applied to each group up to some depth D.

We use the leaf nodes of the vocabulary tree as our dictionary of visual words, $\mathcal{N}$. In addition, as in [12], we use statistics from the vocabulary tree to fix the active recognition update strategy used in our experiments. For each node in the tree a TFIDF-like (Term Frequency Inverse Document Frequency) metric is calculated to capture the node's uniqueness, $w_i = \ln(M/M_i)$, where M is the total number of images in the database and M_i is the number images in the database with at least one feature that passes through node i. The weightings for all nodes passed through by the features of a given training image are summed to generate the 'viewpoint weighting' of the image. We use these weightings at each step of the algorithm to select the next best view. We use the number of matches returned from the Hough geometric matching method described in Sec. IV-B to select the best matching pose for each object at a given time-step. We then align the weightings for each object based these best pose guesses. We then select the δ which achieves the best viewpoint weighting score averaged across all objects (disregarding views already visited).

C. Parameter setting

Our threshold for recognition τ, as in Sec. III is set to 0.8. We set p_a and p_b in Sec. IV-A to 1 and 2 respectively, similarly to [12]. The parameter p_c in Sec. IV-B is set to 0.7 so that we see at least 2 viewpoints before reaching τ and making a decision. The parameter p_d in Sec. IV-C is set to 0.9, which corresponds roughly to the inverse of the proportion of occluded pixels in the test images, and $p_e = 1/(N|\mathcal{X}|)$ as discussed. We tested neighboring parameter values, and found the given values to perform best, with robustness across a large range of values.

D. Hough Transform implementation

We implemented the Hough Transform geometric matching method as described in Sec. IV-B, with two minor differences. First, our set of transformations includes not only translations, but also scalings and rotations. We discretize the transformations into 32 bins each for x and y translations, 5 bins for scale and 12 bins for rotation. Second, we form the matched pairs (which each cast a vote) by thresholding the Euclidean distances between SIFT descriptors (as in [10]), rather than identifying features associated with the same visual word from the vocabulary tree. This alleviates quantization effects that may be introduced by using matched code-words.

E. Results

We first tested the independent feature model with algorithm A, giving a recognition accuracy of 20%. This low performance was expected given that this model doesn't distinguish between the foreground and the background or occlusions (one object and no poses were identified with algorithm B).

We then tested the following probability models:

- Govender et. al.: This method was presented in [12] which also uses a vocabulary structure to weight object features
- Object Models with binary model (Binary Model A)
- Object Models with occlusion model (Occlusion Model A)
- Object & Viewpoint model with binary model (Binary Model B)
- Object & Viewpoint model with occlusion model (Occlusion Model B)

A confusion matrix was generated for each model. We show the confusion matrix generated for the binary method A in Table I.

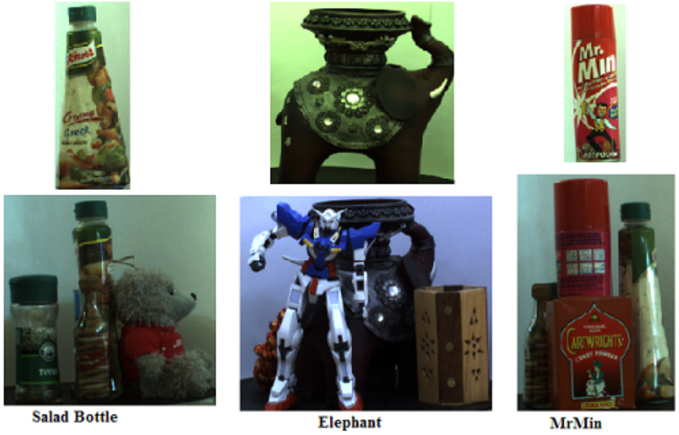

Fig. 1. Examples of the test images used

TABLE I

CONFUSION MATRIX FOR BINARY A MODEL

	Obscured Cereal	Obscured Battery	Obscured Curry box	Obscured Elephant	Obscured Handbag	Obscured MrMin	Obscured Salad Bottle	Obscured Spice Bottle	Obscured Spray Can	Obscured Spray Can 1
Cereal	**0.9800**	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022
Battery	0.0022	**0.9800**	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022
Curry box	0.0021	0.0447	**0.9383**	0.0021	0.0021	0.0021	0.0021	0.0021	0.0021	0.0021
Elephant	0.0022	0.0022	0.0022	**0.9800**	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022
Handbag	0.0022	0.0022	0.0022	0.0022	**0.9800**	0.0022	0.0022	0.0022	0.0022	0.0022
Mr Min	0.0022	0.0022	0.0022	0.0022	0.0022	**0.9800**	0.0022	0.0022	0.0022	0.0022
Salad Bottle	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	**0.9800**	0.0022	0.0022	0.0022
Spice Bottle	0.0412	0.0412	0.8647	0.0020	0.0020	0.0020	0.0020	**0.0412**	0.0020	0.0020
Spray Can	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	0.0022	**0.9800**	0.0022
Spray Can 1	0.0429	0.0020	0.0020	0.0020	0.0020	0.0020	0.9000	0.0020	0.0020	**0.0429**

TABLE II

OBJECT RECOGNITION RESULTS FOR ALL METHODS

	Recognition rate	Sum of Diagonal	Time(s)
Govender et. al. [12]	80%	7.56	404.5
Binary Model A	80%	7.88	448.7
Binary Model B	80%	7.8	1673
Occlusion model A	100%	9.99	240.6
Occlusion model B	100%	10	238.7

TABLE III

OBJECT AND POSE RECOGNITION FOR BINARY AND OCCLUSION B

	Recognition rate	Sum of Diagonal	Pose
Binary Model B	90%	7.8	70%
Occlusion model B	100%	10	90%

The output probabilties for each test object were placed in the respective rows with the diagonal representing the agreement between the true and estimated objects. It achieves a recognition accuracy of 8 out 10 objects. The recognition results for all the methods are presented in Table II.

Occlusion models 1 and 2 recognise all the objects correctly in this challenging dataset. The binary models, along with the method of [12] only recognise 8 out of the 10 objects correctly. Both methods fail to recognise the same objects i.e. the spice bottle and one of the spray can objects. Both these objects are relatively small with fewer features and are significantly occluded in the test images.

We also ran experiments using the binary and occlusion models to recognise the object as well the pose of the test images (to within 20°). The results are displayed below in Table III.

The occlusion model produces the best results. It correctly recognises all objects in the database and is also the fastest method. Given that the occlusion model recognises all the objects in the database, we can conclude that it is important to take into account the object as well as the background features, as well as explicitly model the geometric transformation and occlusion of features. Pose estimation accuracy refers to the system accurately predicting the correct pose of the test objects. Again, the occlusion model performs best, identifying 9 poses to within 20°.

VI. CONCLUSIONS

We presented several Bayesian approaches to active object recognition which allowed information across several viewpoints to be intergrated in a principled manner and provide a quantatative value as to the system's confidence in the object's identity and pose. Our test set consisted of ten objects appearing in cluttered environments with occlusion. The vocabulary tree data structure used to generate the feature statistics can easily incorporate more objects with little or no additional computational complexity. The probabilistic object and viewpoint models created were explicitly designed to cope with such a difficult enviroment. All the models presented achieve excellent results given the challenging dataset. We have shown that using a model which incorporates transformation and occlusion latent variables, as well as incorporating both object and background distributions provides the best result (correctly recognises all the objects) for 3D object recogntion using our method.

VII. Acknowledgments

We would like to thank Deon Sabatta for the use of his vocabulary tree implementation.

References

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

[2] A. Selinger and R. Nelson, "Appearance-based object recognition using multiple views," in *Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 905, 2001.

[3] J. Denzler and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," in *IEEE Transactions on PAMI*, 2002.

[4] E. Sommerlade and I. Reid, "Information-theoretic active scene exploration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[5] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Active object recognition in parametric eigenspace," in *British Machine Vision Conference (BMVC)*, pp. 629–638, 1998.

[6] F. Callari and F. Ferrie, "Active object recognition: Looking for differences," in *International Journal of Computer Vision*, pp. 189–204, 2001.

[7] D. Lowe, "Local feature view clustering for 3d object recognition," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 682–688, 2001.

[8] V. Ferrari, T. Tuytelaars, and L. Gool, "Simultaneous object recognition and segmentation from single or multiple views," in *International Journal of Computer Vision*, vol. 67, pp. 159–188, 2006.

[9] P. Moreels and P. Perona, "Evaluation of feature detectors and descriptors based on 3d objects," in *International Journal of Computer Vision*, vol. 73, pp. 263–284, 2007.

[10] D. Lowe, "Distinctive image features from scale invariant keypoints," in *International Journal of Computer Vision*, vol. 60 of *91-110*, 2004.

[11] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 5, 2006.

[12] N. Govender, J. Claassens, P. Torr, and J. Warrell, "Active object recognition using vocabulary trees," in *IEEE Workshop on Robot Vision*, 2013.

[13] G. Koostra, J. Ypma, and B. de Boer, "Active exploration and keypoint clustering for object recognition," in *IEEE International Conference on Robotics and Automation(ICRA)*, 2008.

[14] Z. Jia, "Active view selection for object and pose recognition," in *International Conference on Computer Vision (ICCV) 3D Object Recognition Workshop*, 2009.

[15] S. A. Hutchinson and A. C. Kak, "Planning sensing strategies in a robot work cell with multi-sensor capabilities," in *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 765–783, 1989.

[16] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "A comparision of probabilistic, possibilistic and evidence theoretic fusion schemes for active object reccognition," in *Computing*, pp. 293–319, 1999.

[17] D. Sabatta, D. Scaramuzza, and R. Siegwart, "Improved appearance-based matching in similar and dynamic environments using a vocabulary tree," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1008 – 1013, 2010.